

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/015085

International filing date: 18 August 2005 (18.08.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2004-244178  
Filing date: 24 August 2004 (24.08.2004)

Date of receipt at the International Bureau: 13 October 2005 (13.10.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 4 年 8 月 2 4 日

出 願 番 号  
Application Number: 特 願 2 0 0 4 - 2 4 4 1 7 8

パリ条約による外国への出願  
に用いる優先権の主張の基礎  
となる出願の国コードと出願  
番号

The country code and number  
of your priority application,  
to be used for filing abroad  
under the Paris Convention, is

J P 2 0 0 4 - 2 4 4 1 7 8

出 願 人  
Applicant(s): 日 本 電 信 電 話 株 式 有 限 公 司

2 0 0 5 年 9 月 2 8 日

特許庁長官  
Commissioner,  
Japan Patent Office

中 嶋



【書類名】 特許願  
【整理番号】 NTTH165607  
【提出日】 平成16年 8月24日  
【あて先】 特許庁長官殿  
【国際特許分類】 G09C 1/00  
【発明者】  
    【住所又は居所】 東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内  
    【氏名】 堀田 英一  
【発明者】  
    【住所又は居所】 東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内  
    【氏名】 小野 諭  
【発明者】  
    【住所又は居所】 東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内  
    【氏名】 石本 英隆  
【発明者】  
    【住所又は居所】 東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内  
    【氏名】 田倉 昭  
【特許出願人】  
    【識別番号】 000004226  
    【氏名又は名称】 日本電信電話株式会社  
【代理人】  
    【識別番号】 100083806  
    【弁理士】  
    【氏名又は名称】 三好 秀和  
    【電話番号】 03-3504-3075  
【手数料の表示】  
    【予納台帳番号】 001982  
    【納付金額】 16,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9701396

## 【書類名】特許請求の範囲

### 【請求項1】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能なイベント順序証明装置が、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明方法であって、

前記利用者装置から前記要求を受信する順序証明要求受信ステップと、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算ステップと、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約ステップと、

前記順序証明要求集約ステップで生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶ステップと、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに複数の前記登録点に関する情報を前記記憶部に記憶する登録点記憶ステップと、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の過去のすべての登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成ステップと、

作成された証明応答を前記利用者装置に送信する証明応答送信ステップと、  
を有することを特徴とするイベント順序証明方法。

### 【請求項2】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能なイベント順序証明装置が、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明方法であって、

前記利用者装置から前記要求を受信する順序証明要求受信ステップと、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算ステップと、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約ステップと、

前記順序証明要求集約ステップで生成される順次集約木に関する情報を記憶部に記憶す

る順次集約木記憶ステップと、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶ステップと、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成ステップと、

作成された証明応答を前記利用者装置に送信する証明応答送信ステップと、  
を有することを特徴とするイベント順序証明方法。

#### 【請求項3】

前記記憶部に記憶される順次集約木に関する情報は、前記順次集約木において割当処理がされた各ノードの位置及び割当値であることを特徴とする請求項1又は2記載のイベント順序証明方法。

#### 【請求項4】

前記記憶部はスタック構造を有し、  
第1のスタックは、前記新登録点の即時補完情報を記憶し、  
前記利用者装置ごとに具備する第2のスタックは、該利用者装置の遅延補完情報を記憶し

さらに前記記憶部は前記利用者装置ごとに直前の登録点を記憶することを特徴とする請求項2記載のイベント順序証明方法。

#### 【請求項5】

前記一定時間間隔終了後に前記順次集約木のルート値を電子的に公表する電子的情報公表ステップを有することを特徴とする請求項1乃至4のいずれか1項に記載のイベント順序証明方法。

#### 【請求項6】

前記利用者装置は、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段を有することを特徴とする請求項1乃至5のいずれか1項に記載のイベント順序証明方法。

#### 【請求項7】

前記順序証明要求集約ステップは、前記一定時間間隔終了後に順次集約木のルート値を、次の順次集約木のリーフに割り当てられた新たな登録点の即時補完情報となるように、次の順次集約木のリーフに割り当てることを特徴とする請求項1乃至6のいずれか1項に記載のイベント順序証明方法。

#### 【請求項8】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前

記要求に応じて複数の証明書を作成するイベント順序証明装置であって、

前記利用者装置から前記要求を受信する順序証明要求受信手段と、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、

前記順序証明要求集約手段生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに複数の前記登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の過去のすべての登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、

作成された証明応答を前記利用者装置に送信する証明応答送信手段と、  
を有することを特徴とするイベント順序証明装置。

#### 【請求項9】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置であって、

前記利用者装置から前記要求を受信する順序証明要求受信手段と、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、

前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定

義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、

作成された証明応答を前記利用者装置に送信する証明応答送信手段と、  
を有することを特徴とするイベント順序証明装置。

【請求項10】

前記記憶部に記憶される順次集約木に関する情報は、前記順次集約木において割当処理がされた各ノードの位置及び割当値であることを特徴とする請求項8又は9記載のイベント順序証明装置。

【請求項11】

前記記憶部はスタック構造を有し、  
第1のスタックは、前記新登録点の即時補完情報を記憶し、  
前記利用者装置ごとに具備する第2のスタックは、該利用者装置の遅延補完情報を記憶し、  
さらに前記記憶部は前記利用者装置ごとに直前の登録点を記憶することを特徴とする請求項9記載のイベント順序証明装置。

【請求項12】

前記一定時間間隔終了後に前記順次集約木のルート値を電子的に公表する電子的情報公表ステップを有することを特徴とする請求項8乃至11のいずれか1項に記載のイベント順序証明装置。

【請求項13】

前記利用者装置は、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段を有することを特徴とする請求項8乃至12のいずれか1項に記載のイベント順序証明装置。

【請求項14】

前記順序証明要求集約手段は、前記一定時間間隔終了後に順次集約木のルート値を、次の順次集約木のリーフに割り当てられた新たな登録点の即時補完情報となるように、次の順次集約木のリーフに割り当てることを特徴とする請求項8乃至13のいずれか1項に記載のイベント順序証明装置。

【請求項15】

請求項1乃至7のいずれか1項に記載のイベント順序証明方法の各ステップを前記イベント順序証明装置に実行させることを特徴とするイベント順序証明プログラム。

【請求項16】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と、該利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置と、  
を備えるイベント順序証明システムにおいて前記証明書の正当性を検証するイベント順序証明検証プログラムであって、

前記イベント順序証明装置は、

前記利用者装置から前記要求を受信する順序証明要求受信手段と、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、

前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに複数の前記登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の過去のすべての登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、

作成された証明応答を前記利用者装置に送信する証明応答送信手段と、

を有し、

前記利用者装置は、

前記要求を前記イベント順序証明装置に送信する要求送信手段と、

前記イベント順序証明装置から前記要求に対する前記証明応答を受信する証明応答受信手段と、

受信した証明応答を記憶する証明応答記憶手段と、

受信し記憶した複数の前記証明書のうち、検証対象の証明書を検証するコンピュータに送信する検証要求手段と、

前記コンピュータから検証結果を受信する検証結果受信手段と、

を有し、

2つの前記利用者装置から検証するそれぞれの1つの前記証明書を受信するか、或いは1つの前記利用者装置から検証する2つの前記証明書を受信する証明書受信ステップと、

受信した各証明書の順次集約木特定情報に基づいて、発行された順序が時間的に前であると判断された証明書を第1の証明書、後であると判断された証明書を第2の証明書とすると、前記第1の証明書を送信した利用者装置に、前記第2の証明書の順次集約木特定情報を送信する順次集約木特定情報送信ステップと、

前記第1の証明書を送信した利用者装置から、前記第1の証明書の前記第2の証明書の発行以降の登録点における遅延補完情報を受信する遅延補完情報受信ステップと、

前記順次集約木の特定のノードに対して、前記第2の証明書に含まれる該ノードの割当値と、前記第1の証明書および前記遅延補完情報から計算された該ノードの割当値が一致するか否かの検証に基づいて、前記各証明書の正当性及び前記第1の証明書の登録点が前



記第2の証明書の登録点より時間的に前であることを証明する検証ステップと、

検証した結果を前記2つの或いは1つの利用者装置に送信する検証結果送信ステップと

、前記コンピュータに実行させることを特徴とするイベント順序証明検証プログラム。

【請求項17】

前記利用者装置は、

前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算可能でないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段、

を有し、

前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、上記利用者装置により上記利用者サイド電子的情報公表手段で割当値が公表されたノードの割当値と、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータから該ノード割当値が計算できるときには計算された該ノードの割当値が一致するか否かにより、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータが改ざんされていないことの検証を行う利用者サイド公表値による検証ステップを前記コンピュータに実行させることを特徴とする請求項16記載のイベント順序証明検証プログラム。

【請求項18】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と、該利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置と、を備えるイベント順序証明システムにおいて前記証明書の正当性を検証するイベント順序証明検証プログラムであって、

前記イベント順序証明装置は、

前記利用者装置から前記要求を受信する順序証明要求受信手段と、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、

前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記

憶手段と、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、

作成された証明応答を前記利用者装置に送信する証明応答送信手段と、

を有し、

前記利用者装置の各登録点のうち、前記順次集約木の最も右に割り付けられた登録点を暫定終端点とし、該暫定終端点の割付処理が終了した時点において、所定の登録点の取得可能な補完情報すべてを計算することを、前記所定の登録点の証明書に対するインクリメンタル完全化と定義すると、

前記利用者装置は、

前記要求を前記イベント順序証明装置に送信する要求送信手段と、

前記イベント順序証明装置から前記要求に対する前記証明応答を受信する証明応答受信手段と、

受信した証明応答を記憶する証明応答記憶手段と、

受信し記憶した複数の証明応答のうち、検証対象の証明書に対して前記インクリメンタル完全化の処理を行うインクリメンタル完全化手段と、

インクリメンタル完全化された検証対象の証明書を検証するコンピュータに送信する検証要求手段と、

前記コンピュータから検証結果を受信する検証結果受信手段と、

を有し、

2つの前記利用者装置から検証するそれぞれの1つの前記証明書を受信するか、或いは1つの前記利用者装置から検証する2つの前記証明書を受信する証明書受信ステップと、

受信した各証明書の順次集約木特定情報に基づいて、発行された順序が時間的に前であると判断された証明書を第1の証明書、後であると判断された証明書を第2の証明書とすると、前記第1の証明書を送信した利用者装置に、前記第2の証明書の順次集約木特定情報を送信する順次集約木特定情報送信ステップと、

前記第1の証明書を送信した利用者装置から、前記第1の証明書の前記第2の証明書の発行以降の登録点における遅延補完情報を受信する遅延補完情報受信ステップと、

前記順次集約木の特定のノードに対して、前記第2の証明書に含まれる該ノードの割当値と、前記第1の証明書および前記遅延補完情報から計算された該ノードの割当値が一致するか否かの検証に基づいて、前記各証明書の正当性及び前記第1の証明書の登録点が前記第2の証明書の登録点より時間的に前であることを証明する検証ステップと、

検証した結果を前記2つの或いは1つの利用者装置に送信する検証結果送信ステップと

を前記コンピュータに実行させることを特徴とするイベント順序証明検証プログラム。

#### 【請求項19】

前記利用者装置は、

前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算可能でないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段、

を有し、

前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、上記利用者装置により上記利用者サイド電子的情報公表手段で割当値が公表されたノードの割当値と、前記証明

書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータから該ノード割当値が計算できるときには計算された該ノードの割当値が一致するか否かにより、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータが改ざんされていないことの検証を行う利用者サイド公表値による検証ステップを前記コンピュータに実行させることを特徴とする請求項18記載のイベント順序証明検証プログラム。

#### 【請求項20】

所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と、該利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置とを備えるイベント順序証明システムにおける前記利用者装置のイベント順序証明検証プログラムであって、

前記イベント順序証明装置は、

前記利用者装置から前記要求を受信する順序証明要求受信手段と、

前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、

前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、

前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、

前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、

前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、

前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、

作成された証明応答を前記利用者装置に送信する証明応答送信手段と、

を有し、

前記利用者装置の各登録点のうち、前記順次集約木の最も右に割り付けられた登録点を暫定終端点とし、該暫定終端点の割付処理が終了した時点において、所定の登録点の取得可能な補完情報すべてを計算することを、前記所定の登録点の証明書に対するインクリメンタル完全化と定義すると、

前記要求を前記イベント順序証明装置に送信する要求送信ステップと、

前記イベント順序証明装置から前記要求に対する前記証明応答を受信する証明応答受信ステップと、

受信した証明応答を記憶する証明応答記憶ステップと、

受信し記憶した複数の証明応答のうち、検証対象の証明書に対して前記インクリメンタル完全化の処理を行うインクリメンタル完全化ステップと、  
を前記利用者装置に実行させることを特徴とするイベント順序証明検証プログラム。

【請求項 21】

前記インクリメンタル完全化は、利用者装置がイベント順序証明装置から受信し記憶した証明応答を用いて、木構造を構成することなく実行されることを特徴とする  
請求項 18 乃至 20 のいずれか 1 項に記載のイベント順序証明検証プログラム。

【請求項 22】

前記インクリメンタル完全化は、  
所定の登録点の、前記暫定終端点の割付処理が終了した時点において取得可能な補完情報の各要素に対して、  
前記利用者装置がイベント順序証明装置から受信し記憶した 1 つあるいは複数の証明応答の中から  
該要素を直接含むかあるいは該要素を計算するために十分な情報含むような 1 つの証明応答を選出し、  
該証明応答から該要素を計算することにより、  
木構造を構成することなく実行されることを特徴とする  
請求項 21 に記載のイベント順序証明検証プログラム。

【請求項 23】

前記インクリメンタル完全化は、  
前記利用者装置の前記暫定終端点より左に位置するすべての登録点に対して行うことを特徴とする請求項 21 記載のイベント順序証明検証プログラム。

【請求項 24】

前記暫定終端点の、左に位置する前記利用者装置の登録点  $a2$  とその左に位置し  $a2$  に最も近い該利用者装置のもう一つの登録点を  $a1$  について、  
 $a1$  の該暫定終端点の割付処理が終了した時点において取得可能な補完情報の全てと、  
 $a2$  における受理証明書から、 $a2$  の該暫定終端点の割付処理が終了した時点において取得可能な補完情報の全てを計算することを完全化波及処理と定義すると、

前記インクリメンタル完全化は、

前記暫定終端点の、左に位置し、該暫定終端点に最も近い、前記利用者装置の登録点  $a$  に対して該暫定終端点の割付処理が終了した時点において取得可能な補完情報すべてを前記利用者装置が受信し記憶した証明応答から取得あるいは計算することから始まり、  
暫定終端点の左に位置する各登録点の該暫定終端点の割付処理が終了した時点において取得可能な補完情報すべてを計算する処理を、このような登録点のうち一番右に位置する上記登録点  $a$  からはじめ、上記完全化波及処理を用いて、順次その左に位置する登録点に対して実行することにより木構造を構成することなく実行されることを特徴とする請求項 23 に記載のイベント順序証明検証プログラム。

【請求項 25】

前記インクリメンタル完全化は、前記暫定終端点までの各登録点を適宜抽出し、この抽出された登録点間の局所領域に分割し、分割された各局所領域において最も右に割り付けられた登録点を暫定終端点と仮定して、インクリメンタル完全化を行うとともに、抽出された各登録点の取得可能な補完情報すべてを計算する方法により行われることを特徴とする請求項 18 乃至 24 のいずれか 1 項に記載のイベント順序証明検証プログラム。

【請求項 26】

前記イベント順序証明装置は、

前記一定時間間隔終了後に前記順次集約木のルート値を電子的に公表する電子的情報公表手段を有し、

前記一定時間間隔終了後に、前記所定の登録点に関する情報と前記インクリメンタル完全化ステップで計算された補完情報から、前記順次集約木のルート値を計算するルート値計算ステップと、

電子的に公表された前記順次集約木のルート値と計算されたルート値が一致するか否かの検証を行うルート値検証ステップと、  
を前記利用者装置に実行させることを特徴とする請求項 18 乃至 25 のいずれか 1 項に記載のイベント順序証明検証プログラム。

【書類名】明細書

【発明の名称】 イベント順序証明方法、イベント順序証明装置、イベント順序証明プログラム、及びイベント順序証明検証プログラム

【技術分野】

【0001】

本発明は、デジタル・データの生成を伴うイベントの生起順序を証明するイベント順序証明技術に関する。

【背景技術】

【0002】

イベント順序証明技術は、デジタル・データの生成を伴う複数のイベントの間の生起順序を証明するとともに、そのようなイベントに伴って生成されたデジタル・データが何であったかを証明する技術である。

【0003】

近年、インターネット上での電子商取引の活発化や、デジタル文書管理の利用拡大に伴い、「誰が、いつ、どんなデータを生成し、送信したか」を第三者が証明する電子公証の仕組みが必要とされている。電子公証は、送受信者の特定、到達確認、送受信等の前後関係の証明、改ざんの検知、電子文書保管等の機能を具備するものであるが、イベント順序証明技術は、このうち、前後関係の証明及び改ざんの検知の機能を実現するものである。

【0004】

図53は、このようなイベント順序証明技術を用いたイベント順序証明システムを説明する図である。同図に示すイベント順序証明システム900は、利用者（要求者、検証者）30がイベント順序証明の対象データをイベント順序証明装置10に送信すると、イベント順序証明装置10が利用者30から要求された対象データに対して受付の順番を示すデータを付したイベント順序受理証明書を生成し、該イベント順序受理証明書を利用者30に返信するようになっている。

【0005】

一般に、イベントの順序証明を行うシステムに対しては、次の要件がある。

【0006】

第1には、イベント順序証明装置10が、全面的には信頼できないとしても、システム全体としては、高い安全性を確保できることである。

【0007】

第2には、デジタル・データの出力を伴うようなイベントを生成する情報処理機関が一定期間に取得するイベント順序証明からなるシーケンスについて、該シーケンスに属する任意の2つのイベント順序証明の時間的順序を証明できることである。

【0008】

第3には、デジタル・データの出力を伴うようなイベントを生成する2つの情報処理機関甲と乙が一定期間に実行する複数のイベント順序証明要求の応答であるイベント順序証明からなる2つのシーケンスについて、これら2つのシーケンスを合併した結果である合併シーケンスに属する任意の2つのイベント順序証明の時間的順序の証明を、甲と乙が、この二者の間で特別な同期手段を講じることなく、実行できることである。

【0009】

第4には、デジタル・データの出力を伴うようなイベントを生成するイベント順序証明システムの2つの利用者装置甲と乙の各々が取得する2つのイベント順序証明のシーケンスに対して、所定の複数の順序証明要求を所定の方法で集約した集約値を甲のイベント順序証明のシーケンスに基づいて計算すること、及び乙のイベント順序証明のシーケンスに基づいて計算すること、及びこの2つの計算結果が一致することの検証を可能とすることである。このことは、甲の取得するイベント順序証明のシーケンスと乙の取得するイベント順序証明のシーケンスが整合していることを検証するために必要である。

【0010】

第5には、上記第1及至第4の課題として述べた機能を持つ順序証明システムを、システ

ムを構成する各装置の資源と性能、及び装置間を結ぶネットワークの資源と性能についての種々の条件において、実用的な実現方法を持つことである。具体的には、木構造等の非循環有向グラフを用いて順序証明を行う場合、上記非循環グラフが計算機のメモリに収まらない場合においても実現できるようにスケラビリティを持って実現するためには、いずれの装置（順序証明装置およびそれを利用する利用者装置）も上記非循環グラフをメモリ上に展開する必要がないような方法で実現することが要求される。また同じくスケラビリティの観点から、当該の非循環グラフが大きくなっても、順序証明装置とその利用者装置の間の通信量が過大とならないことが要求され、装置間の通信量が当該の非循環グラフのノードの数の対数のオーダーで抑えられるならばこの要求は満たされる。一般に、ネットワークで結ばれた計算機システムにより所定の機能を実現するために必要なメモリ量及び装置間の通信量と、個々の装置における処理量はトレードオフの関係にあり、所定の機能を持つシステムをメモリ量、装置の処理能力、およびネットワークの伝送容量等についての種々の状況化で実用可能とするためには、処理量が実用的な範囲でメモリ量および通信量を小さくする実現方式、及び逆にメモリ量および通信量が実用的な範囲で処理量を小さくする実現方式を提供することが有用である。

#### 【0011】

イベント順序証明装置 10で発行されたイベント受理証明書は、PKI (Public Key Infrastructure; 公開鍵基盤) のもとでデジタル署名を偽造防止／証明手段として採用する場合には、一般に、利用者30から送られた対象データに受付の順番を付した署名対象データに対するデジタル署名を含んだイベント順序受理証明書となっている。

#### 【0012】

このように、イベント順序受理証明書にデジタル署名を含め、イベント順序証明の真正性の主要な根拠としてこのデジタル署名を用いるイベント順序証明システムに関しては、イベント順序証明装置 10の不正に対して対抗策がなくサービスの安全性はイベント順序証明装置の信頼に全面的に依存しているため、上記第1の要件を満たす上で問題がある。

上記の問題に対応して、イベント順序証明の真正性の主要な根拠としてはデジタル署名を用いないイベント順序証明の方法も提案されている。例えば、線形リンキング (Linear Linking Protocol) による方法（例えば、非特許文献1および非特許文献2参照。）は、イベント順序証明装置 10が仮に信頼できないとしても、衝突困難一方向関数の衝突困難性を基礎に高い安全性を確保することが可能となっている方法である。図54は、PKIに依存しない線形リンキングによるイベント順序証明システムを説明する図である。同図に示すイベント順序証明システム910は、複数の利用者30のイベント順序証明対象データ（ハッシュ値）を相互に関連付けるリンク情報 $L_N$ を生成し、リンク情報 $L_N$ を含むイベント順序受理証明書を返信するようになっており、各イベント順序受理証明書が、それまでに生成されたすべてのイベント順序受理証明書に依存するようになっている。そして、このリンク情報の一部（ $L_M, L_N$ ）が定期的にマスメディア等（例えば新聞）に公表されるので、これにより、イベント順序証明装置 10の不正を防止し、結果としてシステム全体の信頼を高めることができるようになっている。

#### 【0013】

この線形リンキングの方式については、イベント順序証明装置10の不正を検出するために利用者30相互の協力が必要であるという問題点がある。また、2つのイベント順序受理証明書C1とC2の発行順序を検証するためには、C1が発行されてからC2が発行されるまでの間に為された全てのイベント順序証明要求に関する大量のデータを取得する必要があり、上記第5の要件を満たす上で問題がある。さらに、利用者30が取得したイベント順序受理証明書を公表された情報と照合して検証するため、即ち、該イベント順序受理証明書と公表された情報が衝突困難一方向関数により所定の方式で関係づけられることを検証するためには、利用者30は、イベント順序証明装置10から、上記リンク情報の一部を公表する2つのタイミングに挟まれた期間に受け付けられた全ての順序証明要求の数に比例する大量のデータを取得する必要があり、やはり上記第5の要件を満たす上で問題がある。

この問題を部分的に解決するための方法も提案されている。例えば、非特許文献3および非特許文献4においては、一定期間にイベント順序証明装置10で処理されたイベント順序証明要求をまとめ公表するデータを計算するために、非特許文献1および非特許文献2で使われている線形のリストの代わりに、木構造等の非循環有向グラフを用いることにより、2つのイベント順序受理証明書発行順序を検証するために必要なデータ、および利用者30がイベント順序受理証明書の検証を行うために必要なデータの量を、該一定期間に受け付けられるイベント順序証明要求の数に比例する量から、その対数（底2）に比例する量に著しく削減する方法を提案している。

#### 【0014】

この木構造等の非循環有向グラフを用いる方法は、複数のイベント順序証明を纏める一定の期間が終了し、該纏めたデータを公表し、その期間に発行された各イベント順序証明に対して、該イベント順序証明が上記纏めたデータと衝突困難一方向関数により付けられることを証明するための付加データを、イベント順序証明装置から該イベント順序証明を要求した利用者装置が取得してからのみ、上記の第1及至第4の要件として述べた機能を提供している。従って、上述した従来技術の木構造等の非循環有向グラフを用いる方式では、上記の一定期間が終了するまでは、上記第1及至第4の要件として挙げた機能を提供できないという問題、また一定期間が終了する前に順序証明装置が故障したときにはその故障が回復するまで上記第1及至第4の要件として挙げた機能を提供できないという問題、さらに一定期間が終了する前に順序証明装置が故障し該一定期間に発行した順序証明に関するデータを消失した場合には上記第1及至第4の要件として挙げた機能を永久に提供できなくなるという問題がある。

【非特許文献1】S. Haber and W. Stornetta, How to Time-Stamp a Digital Document, Journal of Cryptology, Vol. 3, No. 2, pp. 99—111, 1991

【非特許文献2】J.-J. Quisquater, H. Massias, J.S. Avila, B. Preneel, B. Van Rompay: Specification and implementation of a timestamping system, Technical Report of Universite Catholique de Louvain, 1999, URL: <http://www.dice.ucl.ac.be/crypto/TIMESEC/TR4.tgz>

【非特許文献3】A. Buldas, P. Laud, H. Lipmaa and J. Villiemson:

Time-stamping with binary linking schemes, in Proceedings of Advances on Cryptology (CRYPTO'98), ed. H. Krawczyk, pp.486—501, Springer-Verlag, 1998

【非特許文献4】A. Buldas, H. Lipmaa and B. Schoenmakers,

Optimally efficient accountable time-stamping, in Proceedings of Public Key Cryptography 2000 (PKC2000), eds. Y. Zheng and H. Imai, pp.293—305, Springer-Verlag, January 2000

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0015】

本発明は、上記の問題を解決するためになされたものであり、木構造を用い衝突困難一方向関数の衝突困難性を基礎にイベント順序証明を行うシステムにおいて、複数のイベント順序証明を纏める一定の期間が終了する前であっても、上記の第1及至第4の要件として挙げた機能を提供すると同時に、

上記第1及至第4の課題として述べた機能を持つ順序証明システムを、上記木構造が大きくネットワークが木構造の大きさに比例する通信容量を許容しない場合でも実現できるように、いずれの装置も木構造をメモリ上に展開する必要がなく、順序証明装置とその利用者装置の間の通信量が木構造の高さの対数のオーダーで抑えられ、かつ各装置における処理量が実用的であるような実現方式を提供すると同時に、ネットワークが木構造の大きさに比例する通信容量を許容する場合には、各装置における処理量が著しく小さくかつ必要メモリ量および装置間の通信量が実用的であるという意味で上記第5の要件を満たすような、イベント順序証明方法、イベント順序証明システムにおける証明装置、及びイベント順



序証明プログラム、及びイベント順序証明検証プログラムを提供することを目的とする。

【課題を解決するための手段】

【0016】

上記目的を達成するため、請求項1記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能なイベント順序証明装置が、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明方法であって、前記利用者装置から前記要求を受信する順序証明要求受信ステップと、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算ステップと、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約ステップと、前記順序証明要求集約ステップで生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶ステップと、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに複数の前記登録点に関する情報を前記記憶部に記憶する登録点記憶ステップと、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の過去のすべての登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成ステップと、作成された証明応答を前記利用者装置に送信する証明応答送信ステップと、を有することを特徴とする。

【0017】

請求項2記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能なイベント順序証明装置が、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明方法であって、前記利用者装置から前記要求を受信する順序証明要求受信ステップと、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算ステップと、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約ステップと、前記順序証明要求集約ステップで生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶ステップと、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で

定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶ステップと、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成ステップと、作成された証明応答を前記利用者装置に送信する証明応答送信ステップと、を有することを特徴とする。

#### 【0018】

請求項3記載の本発明は、請求項1又は2記載の発明において、前記記憶部に記憶される順次集約木に関する情報は、前記順次集約木において割当処理がされた各ノードの位置及び割当値であることを特徴とする。

#### 【0019】

請求項4記載の本発明は、請求項2記載の発明において、前記記憶部はスタック構造を有し、第1のスタックは、前記新登録点の即時補完情報を記憶し、前記利用者装置ごとに具備する第2のスタックは、該利用者装置の遅延補完情報を記憶し、さらに前記記憶部は前記利用者装置ごとに直前の登録点を記憶することを特徴とする。

#### 【0020】

請求項5記載の本発明は、請求項1乃至4のいずれか1項に記載の発明において、前記一定時間間隔終了後に前記順次集約木のルート値を電子的に公表する電子的情報公表ステップを有することを特徴とする。

#### 【0021】

請求項6記載の本発明は、請求項1乃至5のいずれか1項に記載の発明において、前記利用者装置は、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段を有することを特徴とする。

#### 【0022】

請求項7記載の本発明は、請求項1乃至6のいずれか1項に記載の発明において、前記順序証明要求集約ステップは、前記一定時間間隔終了後に順次集約木のルート値を、次の順次集約木のリーフに割り当てられた新たな登録点の即時補完情報となるように、次の順次集約木のリーフに割り当ててことを特徴とする。

#### 【0023】

請求項8記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置であって、前記利用者装置から前記要求を受信する順序証明要求受信手段と、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当ててルート値を計算する順序証明要求集約手段と、前記順序証明要求集約手段生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点か

ら前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに複数の前記登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の過去のすべての登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、作成された証明応答を前記利用者装置に送信する証明応答送信手段と、を有することを特徴とする。

#### 【0024】

請求項9記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置であって、前記利用者装置から前記要求を受信する順序証明要求受信手段と、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、作成された証明応答を前記利用者装置に送信する証明応答送信手段と、を有することを特徴とする。

#### 【0025】

請求項10記載の本発明は、請求項8又は9記載の発明において、前記記憶部に記憶される順次集約木に関する情報は、前記順次集約木において割当処理がされた各ノードの位置及び割当値であることを特徴とする。

#### 【0026】

請求項11記載の本発明は、請求項9記載の発明において、前記記憶部はスタック構造を有し、第1のスタックは、前記新登録点の即時補完情報を記憶し、前記利用者装置ごとに具備する第2のスタックは、該利用者装置の遅延補完情報を記憶し、さらに前記記憶部

は前記利用者装置ごとに直前の登録点を記憶することを特徴とする。

【0027】

請求項12記載の本発明は、請求項8乃至11のいずれか1項に記載の発明において、前記一定時間間隔終了後に前記順次集約木のルート値を電子的に公表する電子的情報公表ステップを有することを特徴とする。

【0028】

請求項13記載の本発明は、請求項8乃至12のいずれか1項に記載の発明において、前記利用者装置は、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段を有することを特徴とする。

【0029】

請求項14記載の本発明は、請求項8乃至13のいずれか1項に記載の発明において、前記順序証明要求集約手段は、前記一定時間間隔終了後に順次集約木のルート値を、次の順次集約木のリーフに割り当てられた新たな登録点の即時補完情報となるように、次の順次集約木のリーフに割り当てることを特徴とする。

【0030】

請求項15記載の本発明は、請求項1乃至7のいずれか1項に記載のイベント順序証明方法の各ステップを前記イベント順序証明装置に実行させるイベント順序証明プログラムであることを特徴とする。

【0031】

請求項16記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と、該利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置と、を備えるイベント順序証明システムにおいて前記証明書の正当性を検証するイベント順序証明検証プログラムであって、前記イベント順序証明装置は、前記利用者装置から前記要求を受信する順序証明要求受信手段と、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに複数の前記登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の過去のすべての登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作

成する証明応答作成手段と、作成された証明応答を前記利用者装置に送信する証明応答送信手段と、を有し、前記利用者装置は、前記要求を前記イベント順序証明装置に送信する要求送信手段と、前記イベント順序証明装置から前記要求に対する前記証明応答を受信する証明応答受信手段と、受信した証明応答を記憶する証明応答記憶手段と、受信し記憶した複数の前記証明書のうち、検証対象の証明書を検証するコンピュータに送信する検証要求手段と、前記コンピュータから検証結果を受信する検証結果受信手段と、を有し、2つの前記利用者装置から検証するそれぞれの1つの前記証明書を受信するか、或いは1つの前記利用者装置から検証する2つの前記証明書を受信する証明書受信ステップと、受信した各証明書の順次集約木特定情報に基づいて、発行された順序が時間的に前であると判断された証明書を第1の証明書、後であると判断された証明書を第2の証明書とすると、前記第1の証明書を送信した利用者装置に、前記第2の証明書の順次集約木特定情報を送信する順次集約木特定情報送信ステップと、前記第1の証明書を送信した利用者装置から、前記第1の証明書の前記第2の証明書の発行以降の登録点における遅延補完情報を受信する遅延補完情報受信ステップと、前記順次集約木の特定のノードに対して、前記第2の証明書に含まれる該ノードの割当値と、前記第1の証明書および前記遅延補完情報から計算された該ノードの割当値が一致するか否かの検証に基づいて、前記各証明書の正当性及び前記第1の証明書の登録点が前記第2の証明書の登録点より時間的に前であることを証明する検証ステップと、検証した結果を前記2つの或いは1つの利用者装置に送信する検証結果送信ステップと、を前記コンピュータに実行させることを特徴とする。

#### 【0032】

ここで、「コンピュータ」とは、検証対象の当事者となっている利用者装置自身も含む意味である。尚、以下の発明においても同様である。

#### 【0033】

請求項17記載の本発明は、請求項16記載の発明において、前記利用者装置は、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段を有し、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、上記利用者装置により上記利用者サイド電子的情報公表手段で割当値が公表されたノードの割当値と、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータから該ノード割当値が計算できるときには計算された該ノードの割当値が一致するか否かにより、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータが改ざんされていないことの検証を行う利用者サイド公表値による検証ステップを前記コンピュータに実行させることを特徴とする。

請求項18記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と、該利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置と、を備えるイベント順序証明システムにおいて前記証明書の正当性を検証するイベント順序証明検証プログラムであって、前記イベント順序証明装置は、前記利用者装置から前記要求を受信する順序証明要求受信手段と、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、前記順序証明要求集約手段で生成される順次集約木に関

する情報を記憶部に記憶する順次集約木記憶手段と、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、作成された証明応答を前記利用者装置に送信する証明応答送信手段と、を有し、前記利用者装置の各登録点のうち、前記順次集約木の最も右に割り付けられた登録点を暫定終端点とし、該暫定終端点の割付処理が終了した時点において、所定の登録点の取得可能な補完情報すべてを計算することを、前記所定の登録点の証明書に対するインクリメンタル完全化と定義すると、前記利用者装置は、前記要求を前記イベント順序証明装置に送信する要求送信手段と、前記イベント順序証明装置から前記要求に対する前記証明応答を受信する証明応答受信手段と、受信した証明応答を記憶する証明応答記憶手段と、受信し記憶した複数の証明応答のうち、検証対象の証明書に対して前記インクリメンタル完全化の処理を行うインクリメンタル完全化手段と、インクリメンタル完全化された検証対象の証明書を検証するコンピュータに送信する検証要求手段と、前記コンピュータから検証結果を受信する検証結果受信手段と、を有し、2つの前記利用者装置から検証するそれぞれの1つの前記証明書を受信するか、或いは1つの前記利用者装置から検証する2つの前記証明書を受信する証明書受信ステップと、受信した各証明書の順次集約木特定情報に基づいて、発行された順序が時間的に前であると判断された証明書を第1の証明書、後であると判断された証明書を第2の証明書とすると、前記第1の証明書を送信した利用者装置に、前記第2の証明書の順次集約木特定情報を送信する順次集約木特定情報送信ステップと、前記第1の証明書を送信した利用者装置から、前記第1の証明書の前記第2の証明書の発行以降の登録点における遅延補完情報を受信する遅延補完情報受信ステップと、前記順次集約木の特定のノードに対して、前記第2の証明書に含まれる該ノードの割当値と、前記第1の証明書および前記遅延補完情報から計算された該ノードの割当値が一致するか否かの検証に基づいて、前記各証明書の正当性及び前記第1の証明書の登録点の前記第2の証明書の登録点より時間的に前であることを証明する検証ステップと、検証した結果を前記2つの或いは1つの利用者装置に送信する検証結果送信ステップと、を前記コンピュータに実行させることを特徴とする。

#### 【0034】

請求項19記載の本発明は、請求項18記載の発明において、前記利用者装置は、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、イベント順序証明装置の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段、を有し、前記一定時間間隔終了前に、前記イベント順序証明装置が運用を中断、あるいは前記順次集約木のルート値を計算するのに必要なデータを消失したとき、上記利用者装置により上記利用者サイド電子的情報公表手段で割当値が公表されたノードの割当値と、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータから該ノード割当値が計算できるときには計算された該ノードの割当

値が一致するか否かにより、前記証明書受信ステップで受信したデータと前記遅延補完情報受信ステップで受信したデータが改ざんされていないことの検証を行う利用者サイド公表値による検証ステップを前記コンピュータに実行させることを特徴とする。

請求項20記載の本発明は、所定のデジタル情報の生成を伴うイベントに対して時間的順序の証明を要求する利用者装置と、該利用者装置と通信ネットワークを介して相互に通信可能であり、前記利用者装置からの複数の前記要求に応じて複数の証明書を作成するイベント順序証明装置とを備えるイベント順序証明システムにおける前記利用者装置のイベント順序証明検証プログラムであって、前記イベント順序証明装置は、前記利用者装置から前記要求を受信する順序証明要求受信手段と、前記証明要求に含まれるデジタル情報から予め定めた手順に従って順次割当データを作成する順次割当データ計算手段と、前記順次割当データを時刻順に有向木のリーフに左から順次割り当て、一定時間間隔ごとに1つの集約木が生成される順次集約木において、同一の親を有する複数の子に割り当てられたそれぞれの割当値を接続した接続値に所定の衝突困難一方向関数を適用した結果値を前記親の割当値とする計算方法により、計算可能なノードの割当値を計算するとともに、前記一定時間間隔終了後に前記順次集約木のルートに割り当てるルート値を計算する順序証明要求集約手段と、前記順序証明要求集約手段で生成される順次集約木に関する情報を記憶部に記憶する順次集約木記憶手段と、前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを登録点と定義し、該登録点から前記ルート値を計算するのに必要な他のノードに関する情報を前記登録点の補完情報と定義し、該補完情報のうち、前記順次割当データを前記順次集約木に割り当てた時点において取得可能な補完情報を即時補完情報、前記順次割当データを前記順次集約木に割り当てた時点以後において取得可能な補完情報を遅延補完情報と定義し、リーフa1より右に位置するリーフa2の割当処理が終了した時点で定まる前記リーフa1の遅延補完情報を、リーフa1のリーフa2における遅延補完情報といい、さらに、最新の前記要求から作成された順次割当データが割り当てられた前記順次集約木のリーフを新登録点とすると、前記利用者装置ごとに前記直前の登録点に関する情報を前記記憶部に記憶する登録点記憶手段と、前記利用者装置の新登録点に対して、該新登録点の順次割当データ、該順次割当データが割り当てられた順次集約木及び該順次集約木のリーフを特定する順次集約木特定情報、並びに新登録点の即時補完情報を備える前記証明書と、前記利用者装置の直前の登録点の新登録点における遅延補完情報と、を併せた証明応答を、前記記憶部に記憶された情報から作成する証明応答作成手段と、作成された証明応答を前記利用者装置に送信する証明応答送信手段と、を有し、前記利用者装置の各登録点のうち、前記順次集約木の最も右に割り付けられた登録点を暫定終端点とし、該暫定終端点の割付処理が終了した時点において、所定の登録点の取得可能な補完情報すべてを計算することを、前記所定の登録点の証明書に対するインクリメンタル完全化と定義すると、前記要求を前記イベント順序証明装置に送信する要求送信ステップと、前記イベント順序証明装置から前記要求に対する前記証明応答を受信する証明応答受信ステップと、受信した証明応答を記憶する証明応答記憶ステップと、受信し記憶した複数の証明応答のうち、検証対象の証明書に対して前記インクリメンタル完全化の処理を行うインクリメンタル完全化ステップと、を前記利用者装置に実行させることを特徴とする。

#### 【0035】

請求項21記載の本発明は、請求項18乃至20のいずれか1項に記載の発明において、前記インクリメンタル完全化は、利用者装置がイベント順序証明装置から受信し記憶した証明応答を用いて、木構造を構成することなく実行されることを特徴とする。

#### 【0036】

請求項22記載の本発明は、請求項21に記載の発明において、前記インクリメンタル完全化は、所定の登録点の、前記暫定終端点の割付処理が終了した時点において取得可能な補完情報の各要素に対して、前記利用者装置がイベント順序証明装置から受信し記憶した1つあるいは複数の証明応答の中から該要素を直接含むかあるいは該要素を計算するために十分な情報含むような1つの証明応答を選出し、該証明応答から該要素を計算することにより、木構造を構成することなく実行されることを特徴とする。



#### 【0037】

請求項23記載の本発明は、請求項21記載の発明において、前記インクリメンタル完全化は、前記利用者装置の前記暫定終端点より左に位置するすべての登録点に対して行うことを特徴とする。

#### 【0038】

請求項24記載の本発明は、請求項23に記載の発明において、前記暫定終端点の、左に位置する前記利用者装置の登録点a2とその左に位置しa2に最も近い該利用者装置のもう一つの登録点をa1について、a1の該暫定終端点の割付処理が終了した時点において取得可能な補完情報の全てと、a2における受理証明書から、a2の該暫定終端点の割付処理が終了した時点において取得可能な補完情報の全てを計算することを完全化波及処理と定義すると、前記インクリメンタル完全化は、前記暫定終端点の、左に位置し、該暫定終端点に最も近い、前記利用者装置の登録点aに対して該暫定終端点の割付処理が終了した時点において取得可能な補完情報すべてを前記利用者装置が受信し記憶した証明応答から取得あるいは計算することから始まり、暫定終端点の左に位置する各登録点の該暫定終端点の割付処理が終了した時点において取得可能な補完情報すべてを計算する処理を、このような登録点のうち一番右に位置する上記登録点aからはじめ、上記完全化波及処理を用いて、順次その左に位置する登録点に対して実行することにより木構造を構成することなく実行されることを特徴とする。

#### 【0039】

請求項25記載の本発明は、請求項18乃至24のいずれか1項に記載の発明において、前記インクリメンタル完全化は、前記暫定終端点までの各登録点を適宜抽出し、この抽出された登録点間の局所領域に分割し、分割された各局所領域において最も右に割り付けられた登録点を暫定終端点と仮定して、インクリメンタル完全化を行うとともに、抽出された各登録点の取得可能な補完情報すべてを計算する方法により行われることを特徴とする。

#### 【0040】

請求項26記載の本発明は、請求項18乃至25のいずれか1項に記載の発明において、前記イベント順序証明装置は、前記一定時間間隔終了後に前記順次集約木のルート値を電子的に公表する電子的情報公表手段を有し、前記一定時間間隔終了後に、前記所定の登録点に関する情報と前記インクリメンタル完全化ステップで計算された補完情報から、前記順次集約木のルート値を計算するルート値計算ステップと、電子的に公表された前記順次集約木のルート値と計算されたルート値が一致するか否かの検証を行うルート値検証ステップと、を前記利用者装置に実行させることを特徴とする。

#### 【発明の効果】

#### 【0041】

本発明によれば、木構造を用いてイベント順序を証明するイベント順序証明システムにおいて、イベント順序証明要求をまとめた公表データを用いなくてもまたイベント順序証明機関がイベント順序証明要求をまとめた公表データを公表する前に障害などのためにサービスを中断したときでも、イベント順序証明機関から発行されたイベント順序受理証明書の検証を行うことができ、公表期間の途中であってもまたイベント順序証明機関がイベント順序証明要求をまとめた公表データを公表する前に障害などのためにサービスを中断したときでも、利用者間におけるイベント順序受理証明書発行の時間的前後を検証することができる。

#### 【発明を実施するための最良の形態】

#### 【0042】

以下、本発明の実施の形態を図面を用いて説明する。

#### 【0043】

#### <第1の実施の形態>

#### (1-1. システム構成)

図1は、本発明の第1の実施の形態に係るイベント順序証明システム100のシステム構成図である。同図に示すイベント順序証明システム100は、イベント順序証明装置（以下、



証明装置という) 1、複数のイベント順序証明利用者装置(以下、利用者装置という) 21 ( $1=A, B, \dots, N$ )、及び、以上の各装置を相互に接続する、例えば、インターネット網、電話回線網などにより構成されるコンピュータネットワーク3を備えており、証明装置1が利用者装置21からのイベント順序証明要求(以下、証明要求という)に応じて、イベント順序受理証明書(以下、受理証明書という)を含むイベント順序証明応答(以下、証明応答という)を利用者装置21に返信するようになっている。そして、利用者装置21は、証明装置1から受け取ったこの複数の証明応答から受理証明書を検証することができるようになっている。

#### 【0044】

証明装置1は、コンピュータネットワーク3を介して利用者装置21とデータの送受信を行う送受信部11、利用者装置21からの証明要求として送信されたデジタル・データを順次集約木を用いてまとめるイベント順序証明要求集約部12、受理証明書を含む証明応答を作成するイベント順序証明応答作成部13、証明装置1が一定期間に発行した複数の受理証明書の内容を集約したデータに対して高強度デジタル署名し、公表データとする高強度デジタル署名作成部14、高強度デジタル署名を付加された公表データを電子的に公表する電子的情報公表部15、及び受理証明書をはじめとするイベント順序証明に関する情報を記憶する記憶部16を有する構成である。

#### 【0045】

上述したように、イベント順序証明要求集約部12は、順次集約木を用いてイベント順序証明要求をまとめるが、この順次集約木について図2を用いて説明する。図2は、一定期間(例えば1週間など証明装置1が取り纏めデータを公表するサイクル、順次集約期間という)において、利用者装置21からの証明要求に含まれるデジタル・データの全部あるいは一部を、所定の順次割当データ計算手順に従って計算した結果であるデジタル・データ(これを順次割当データと呼ぶ; 例えば、証明要求に含まれるデジタル・データのハッシュ値)を経時的に順次左側から割り当てる値とする順次集約木の一例を示す図である。尚、利用者装置21からの各証明要求が割り当てられた順次集約木のリーフを登録点ともいう。

#### 【0046】

順次集約木の各ノード(リーフを除く)に割り当てられる値の計算方法は、以下の通りである。順次集約木の親の割当値は、左側の子の割当値 $H'$ と右側の子の割当値 $H''$ を接続(ビット列とビット列の結合)し、所定の衝突困難一方向ハッシュ関数 $h$ を適用した結果であるハッシュ値を計算することにより求められるものであり、これを $h(H' || H'')$ と表す。このようにして下位のレベルの割当値から上位のレベルの割当値を計算して、最終的に最上位のレベル(ルート)の割当値(ルート値) $H$ を求める。

#### 【0047】

以下においては、図2に示すように、16のリーフを有する順次集約木の場合について説明する。尚、順次集約木リーフの数や高さは、順次集約期間が終了するまで確定しない。また、順次集約木においては、リーフへの値の割当は左から順次行われ、レベルが0より大きいノード(即ちリーフではないノード)に対する値の割当は、それが可能になったときにインクリメンタルに行われる。従って、図2の同一の縦線上にある複数のノードに対しては、値の割当が同一の処理単位の中でほぼ同時に行われる。

ここで、順次集約木のレベル $j$ 、番号 $i$ のノードを $(j, i)$ 、 $(j, i)$ の割当値を $V(j, i)$ と表して、図2に示す具体例を説明する。

#### 【0048】

今、順次集約木リーフに割り当てるハッシュ値が $V(0, 5)$ であるとき(登録点が $(0, 5)$ であるとき)、このハッシュ値 $V(0, 5)$ からルート値 $H (=V(4, 0))$ を求めるには、 $V(0, 5)$ に $V(0, 4)$ を左から接続して、ハッシュ値 $h1'$ を計算し、該ハッシュ値 $h1'$ に $V(1, 3)$ を右側から接続してハッシュ値 $h2'$ を計算し、該ハッシュ値 $h2'$ に $V(2, 0)$ を左側から接続してハッシュ値 $h3'$ を計算し、さらに該ハッシュ値 $h3'$ に $V(3, 1)$ を右側から接続してハッシュ値 $H (=V(4, 0))$ を計算すればよい。このような手順により $V(0, 5)$ とそれを補完するデータ(こ

ここでは $V(0, 4)$ 、 $V(1, 3)$ 、 $V(2, 0)$ 、 $V(3, 1)$ からルート値 $H$ が計算できるとき、 $V(0, 5)$ はハッシュ関数 $h$ によりルート値 $H$ にリンクするという。また、順次集約木における $V(0, 5)$ の補完データ（以下、順次集約補完データという）は、

$[(V(0, 4), L), (V(1, 3), R), (V(2, 0), L), (V(3, 1), R)]$

となる。ここで、 $L$ 及び $R$ は、各々、2つのデジタル・データを接続する際に左から接続こと、及び右から接続することを表す。

#### 【0049】

イベント順序証明応答作成部13は、図3に示すような受理証明書 $EOC(y)$ を含む証明応答を作成し、利用者装置21に送信するようになっている。同図によれば、受理証明書 $EOC(y)$ は、利用者から送付されたデジタル・データ $y$ 、上述した順次割当データ計算手順によりデジタル・データ $y$ から計算された順次割当データ $z$ 、順次割当データ $z$ が割当てられた順次集約木を一意に特定できる順次集約木番号、順次割当データ $z$ が割当てられた順次集約木リーフを一意に特定できる順次集約木リーフ番号、およびその時点で取得できる順次集約補完データの一部（これを登録点の即時補完データと言う） $SK$ の位置情報及び割当値を含むように構成されている。

#### 【0050】

また、同図によれば、証明応答には、利用者装置21の過去の各登録点の遅延補完データ $TK$ の位置情報及び割当値も含むように構成されている。尚、遅延補完データ $TK$ とは、当該の証明応答発行後に取得できる順次集約補完データをいい、例えば、図2の具体例においては、 $V(0, 5)$ にとって、 $V(2, 0)$ 、 $V(0, 4)$ は即時補完データであるが、 $V(1, 3)$ 、 $V(3, 1)$ は $V(0, 15)$ が割当てられた時点以降に取得可能な遅延補完データである。また、一般に、あるリーフ $a1$ とそれより右に位置するもう一つのリーフ $a2$ において、リーフ $a2$ の割当処理が終了した時点で定まるリーフ $a1$ の遅延補完データを、 $a1$ の $a2$ における遅延補完データという。図2の具体例においては、ノード $(0, 5)$ のノード $(0, 10)$ における遅延補完データは、ノード $(1, 3)$ である。

#### 【0051】

ここで、本実施の形態における証明応答の具体例を図4を用いて説明する。尚、以下においては、本実施の形態における証明応答の形式を「シーケンス補完方式」と呼ぶ。今、ある利用者装置21からの登録点が $X1$ （ノード $(0, 2)$ ）、 $X2$ （ノード $(0, 11)$ ）、 $X3$ （ノード $(0, 18)$ ）、 $X4$ （ノード $(0, 21)$ ）、 $X5$ （ノード $(0, 29)$ ）、 $X6$ （ノード $(0, 31)$ ）であるとする。

#### 【0052】

シーケンス補完の方式においては各登録点において以下のようなデータが利用者装置21に返されるようになっている。

#### 【0053】

(1)  $X1$ 点における証明応答には、 $X1$ 点の即時補完データが返される（具体的には、ノード $(1, 0)$ の割当値）。

#### 【0054】

(2)  $X2$ 点における証明応答には、 $X2$ 点の即時補完データ、及び $X1$ 点の $X2$ 点における遅延補完データが返される（具体的には、 $X2$ 点の即時補完データとして、ノード $(3, 0)$ 、ノード $(1, 4)$ 、ノード $(0, 10)$ の割当値、 $X1$ 点の $X2$ 点における遅延補完データとして、ノード $(0, 3)$ 、ノード $(2, 1)$ の割当値）。

#### 【0055】

(3)  $X3$ 点における証明応答には、 $X3$ 点の即時補完データ、並びに $X1$ 点及び $X2$ 点の $X3$ 点における遅延補完データが返される（具体的には、 $X3$ 点の即時補完データとして、ノード $(4, 0)$ 、ノード $(1, 8)$ の割当値、 $X1$ 点の $X3$ 点における遅延補完データとして、ノード $(0, 3)$ 、ノード $(2, 1)$ 、ノード $(3, 1)$ の割当値、 $X2$ 点の $X3$ 点における遅延補完データとして、ノード $(2, 3)$ の割当値）。

#### 【0056】

(4)  $X4$ 点における証明応答には、 $X4$ 点の即時補完データ、並びに $X1$ 点、 $X2$ 点及び $X3$ 点の

X4点における遅延補完データが返される（具体的には、X4点の即時補完データとして、ノード(4,0)、ノード(2,4)、ノード(0,20)の割当値、X1点のX4点における遅延補完データとして、ノード(0,3)、ノード(2,1)、ノード(3,1)の割当値、X2点のX4点における遅延補完データとして、ノード(2,3)の割当値、X3点のX4点における遅延補完データとして、ノード(0,19)の割当値）。

#### 【0057】

以下、X5及びX6においても同様である。このように、シーケンス補完方式においては、証明応答として、ある登録点に関して、該登録点の即時補完データ及びある登録点より前に登録された各登録点の該登録点における遅延補完データが含まれるようになっている。尚、各証明応答は、利用者装置21ごとに管理されるようになっている。

#### 【0058】

利用者装置21は、コンピュータネットワーク3を介して証明装置1とデータを送受信する送受信部21、所定のデジタル・データを含む証明要求を複数回行うイベント順序証明要求証部22、証明要求に対する証明応答に含まれた受理証明書を検証するイベント順序証明検証部23、受理証明書を含む証明応答をはじめとしてイベント順序証明に関する情報を記憶する記憶部24を有する構成である。

#### 【0059】

ここで、イベント順序証明検証部23は、受理証明書に対して以下の検証機能を備える。

#### 【0060】

第1の検証機能としては、証明装置1のデジタル署名作成部14及び電子的情報公表部15を介して公表される公表情報に、受理証明書に含まれる順次割当データがハッシュ関数を介してリンクすることを検証するものである。具体的には、順次集約木のルート値として公表された値と利用者装置21で計算されたルート値が一致するか否かを検証するものである。

#### 【0061】

第2の検証機能としては、公表情報が公開される前であっても、利用者装置21の後述する動作により、利用者装置21間における受理証明書発行の時間的前後を検証するものである。

#### 【0062】

以下、第2の検証機能を図5を用いて説明するが、その前に、合流点及び認証点の説明をする。

#### 【0063】

ある順次集約木のリーフaに対して、aからルートに至るパスをaのルート・パスと呼び、 $rtPath(a)$ と書く。また、 $rtPath(a)$ に属するノードのルート以外のものの兄弟ノード(sibling node)からなるノードの並びを認証パスと呼び $authPath(a)$ と書く。

#### 【0064】

このとき、ある順次集約木の2つのリーフa1とa2について、a1より右にa2が位置するとき、a1からルートに至るパスとa2からルートに至るパスの合流する点を、a1とa2の合流点と呼び、合流点のレフト・チャイルド（左側の子）をa1のa2による認証点と呼ぶ。例えば、図4において、登録点X1の登録点X2による認証点は(3, 0)の点であり、登録点X2の登録点X3による認証点は(4, 0)の点である。

#### 【0065】

今、2つの利用者装置2A及び2Bが、それぞれ、シーケンス補完方式により、各登録点の証明応答を取得するものとし、図5に示すa、a1、a2、afを利用者装置2Aの登録点、bを利用者装置2Bの登録点とする。尚、afは、暫定終端点と呼ばれるもので、利用者装置2Aの登録点のうち、最も右に位置する登録点である。図5においては、利用者装置2Aの1つの登録点aがあり、それより右に位置する利用者装置2Bの登録点bがあり、さらにその右に位置する利用者装置2Aの登録点afがある。

#### 【0066】

ここで、シーケンス補完方式においては、一般に、登録点aが登録点bより左に位置す

る場合、認証点のラベル（割当値）は登録点bの証明応答（即時補完データ）に含まれており、また、登録点bにおけるイベント順序証明処理が終了した時点以降（例えば、登録点a<sub>1</sub>の時点）においては、登録点aの遅延補完データから計算できるラベルには、認証点のラベルが含まれるので、登録点aの登録点a<sub>1</sub>における遅延補完データから計算される認証点の割当値と、登録点bの即時補完データに含まれる認証点の割当値が一致するか否かを検証することにより、登録点aと登録点bの時間的前後を検証することができる（詳しくは、後述の順次集約木の性質の項目(3)を参照）。

#### 【0067】

従って、図5の場合、登録点a<sub>1</sub>を現時点としたときの現時点順次集約木における、登録点aの登録点bによる認証点oの割当値V(o)が一致すれば、登録点aの登録が登録点bの登録より前に起こったことを客観的に証明することができる。イベント順序証明検証部23は、このことを後述する動作により検証するものである。

#### 【0068】

尚、以上の各装置は、少なくとも演算機能及び制御機能を備えた中央処理装置（CPU：Central Processing Unit）、プログラムやデータを収納する機能を有するRAM(Random Access Memory)等からなる主記憶装置（メモリ）、ハードディスク（HD）等の電源断時にもデータを記憶し続けることができる2次記憶装置を有する電子的な装置から構成されている。このうち、証明装置1のイベント順序証明要求集約部12、イベント順序証明応答作成部13、デジタル署名作成部14及び電子的情報公表部15、並びに利用者装置21のイベント順序証明要求部22及びイベント順序証明検証部23の処理は、上記CPUによる演算制御機能を具体的に示したものに他ならない。また、証明装置1の記憶部16及び利用者装置21の記憶部24は、上記主記憶装置あるいは2次記憶装置の機能を備えたものである。

#### 【0069】

##### （1-2. システム動作）

次に、以上の構成を有するイベント順序証明システム100におけるイベント順序証明方法、およびイベント順序証明検証方法を図6乃至8を用いて説明する。ここで、図6は、1つの順次集約期間において証明装置1が受理証明書を含む及び証明応答を作成する動作を説明するシーケンス図であり、図7及び8は、利用者装置21が受理証明書に対して第2の検証を行う動作を説明するシーケンス図である。

#### 【0070】

まず、図6を参照して、イベント順序証明方法について説明する。

#### 【0071】

利用者装置21が証明装置1にデジタル・データyを含む証明要求を送信すると、証明装置1は送受信部11を介して、該デジタル・データyを含む証明要求を受信する（ステップS10、S20）。

次に、イベント順序証明要求集約部12が、デジタル・データyを入力の一部あるいは全部として順次割当データzを計算し、該順次割当データzを順次集約木リーフに割り当てて、インクリメンタルに順次集約木を構成していくとともに、イベント順序証明応答作成部13は、受理証明書を含む証明応答（シーケンス補完の方式；登録点の即時補完データ及び登録点より前に登録された各登録点の該登録点における遅延補完データ）を作成し、送受信部11を介して利用者装置21に証明応答を送信する（ステップS30、S40、S50）。

#### 【0072】

これにより、利用者装置21は、受理証明書を含む証明応答を取得することができる（ステップS60）。そして、利用者装置21は、このステップS10及びS60の証明要求送信及び証明応答受信を繰り返す。

#### 【0073】

一方、証明装置1では、順次集約のための一定期間（順次集約期間）内においては、上述した証明装置1の動作は繰り返され、順次集約期間が終了すると、電子的情報公表部17は、順次集約木のルート値を計算し、このルート値を電子的に公表する（ステップS70、S80、S90）。

#### 【0074】

次に、図7を参照しながら、イベント順序証明検証方法について説明する。これは、利用者装置21の第2の検証機能に相当するものである。尚、図7は、利用者装置2Aと2B間のデータのやりとりを示すものであるが、利用者装置2Aが利用者装置2Bに後置点判定要求（利用者装置2Aが受け取った受理証明書の登録点より時間的に後の受理証明書に対する順序判定を利用者装置2Bに要求する）を行うものである。

#### 【0075】

まず、利用者装置2Aは、検証したい受理証明書E0C(a)（登録点aにおける受理証明書）を付けて後置点判定要求を利用者装置2Bに送信する（ステップS110）。後置点判定要求を受信した利用者装置2Bは、受け取った受理証明書E0C(a)からリーフ番号n(a)を抽出し、リーフ番号n(a)より大きなリーフ番号を利用者装置2Bの登録点の中から検索する（ステップS120、S130）。リーフ番号n(a)より大きなリーフ番号が利用者装置2Bの登録点にある場合には、該登録点の一つbを選び、そのリーフ番号n(b)を利用者装置2Aに送信する（ステップS140、S150）。これに対して、リーフ番号n(a)より大きなリーフ番号が利用者装置2Bの登録点にない場合には、比較可能データなしの旨のメッセージを利用者装置2Aに送信する（ステップS142）。

#### 【0076】

リーフ番号n(b)を利用者装置2Bから受信した利用者装置2Aは、リーフ番号n(b)より大きなリーフ番号を持つ利用者装置2Aの暫定登録点a<sub>1</sub>を選び、登録点aの暫定終端点a<sub>1</sub>における遅延補完データlateData(a, a<sub>1</sub>)を取得し、利用者装置2Bに送信する（ステップS160、S170、S180、S190）。尚、比較可能データなしの旨のメッセージを利用者装置2Bから受信したときは、検証を終了する（ステップS144）。

#### 【0077】

利用者装置2Aから遅延補完データlateData(a, a<sub>1</sub>)を受信した利用者装置2Bは、リーフ識別番号n(a)とn(b)から登録点aの登録点bに対する認証点oを計算し、受け取った受理証明書E0C(a)と遅延補完データlateData(a, a<sub>1</sub>)から認証点の割当値を計算する（ステップS210）。次に、登録点bの受理証明書E0C(b)に含まれる即時補完データに上記計算で求めた認証点の割当値が含まれているか否かを検証し、認証点の割当値が含まれているときは、登録点aが登録点bより時間的に前に登録されたという判定結果を利用者装置2Aに送信する（ステップS220、S230）。一方、認証点の割当値が含まれていないときは、登録点aが登録点bより時間的に前に登録されたことを証明できず、何らかの不正があるという判定結果を利用者装置2Aに送信する（ステップS220、S240）。

#### 【0078】

この結果、利用者装置2Aは、判定結果を受信し、取得するので、2利用者間において受理証明書発行の時間的前後を検証することができる（ステップS250、S260）。

#### 【0079】

尚、上述したイベント順序証明検証方法は、利用者装置2Aが利用者装置2Bに後置点判定要求行うものであったが、前置点判定要求（利用者装置2Aが受け取った受理証明書の登録点より時間的に前の受理証明書に対する順序判定を利用者装置2Bに要求する）を行うようにしてもよい。図8は、利用者装置2Aが利用者装置2Bに前置点判定要求行う場合の利用者装置2Aと2B間のデータのやりとりを示すシーケンス図である。

#### 【0080】

まず、利用者装置2Aは、検証したい受理証明書E0C(a)（登録点aにおける受理証明書）を付けて前置点判定要求を利用者装置2Bに送信する（ステップS310）。前置点判定要求を受信した利用者装置2Bは、受け取った受理証明書E0C(a)からリーフ番号n(a)を抽出し、リーフ番号n(a)より小さいリーフ番号、及びリーフ番号n(a)より大きいリーフ番号を利用者装置2Bの登録点の中から検索する（ステップS320、S330）。リーフ番号n(a)より小さいリーフ番号及びリーフ番号n(a)より大きいリーフ番号が利用者装置2Bの登録点にある場合には、リーフ番号n(a)より小さいリーフ番号の登録点bを選ぶとともに、リーフ番号n(a)より大きいリーフ番号の暫定登録点b<sub>1</sub>を選ぶ（ステップS340、S350）。尚、該当する登録点

がない場合には、比較可能データなしの旨のメッセージを利用者装置2Aに送信する（ステップS342）。そして、利用者装置2Aは、比較可能データなしの旨のメッセージを利用者装置2Bから受信したときは、検証を終了する（ステップS344）。

#### 【0081】

次いで、利用者装置2Bは、登録点bの暫定登録点 $b_1$ における遅延補完データlateData (b,  $b_1$ )を取得し、リーフ識別番号 $n(a)$ と $n(b)$ から登録点aの登録点bに対する認証点oを計算し、受理証明書EOC(b)と遅延補完データlateData (b,  $b_1$ )から認証点の割当値を計算する（ステップS360, S370）。次に、登録点aの受理証明書EOC(a)に含まれる即時補完データに上記計算で求めた認証点の割当値が含まれているか否かを検証し、認証点の割当値が含まれているときは、登録点aが登録点bより時間的に後に登録されたという判定結果を利用者装置2Aに送信する（ステップS380, S390）。一方、認証点の割当値が含まれていないときは、登録点aが登録点bより時間的に後に登録されたことを証明できず、何らかの不正があるという判定結果を利用者装置2Aに送信する（ステップS380, S400）。

#### 【0082】

この結果、利用者装置2Aは、判定結果を受信し、取得するので、2利用者間において受理証明書発行の時間的前後を検証することができる（ステップS410, S420）。

#### 【0083】

尚、上述したイベント順序証明検証方法は、利用者装置2A及び利用者装置2Bが受理証明書発行の時間的前後を検証したものであったが、本発明はこれに限定されず、当事者以外の第3者機関が検証を行ってもよい。この場合には、利用者装置2A及び利用者装置2Bが検証に必要な情報を第3者機関に送信し、第3者機関が検証を行うものである。

#### 【0084】

従って、第1の実施の形態のイベント順序証明システム100によれば、本構造を用いてイベント順序を証明するイベント順序証明システムにおいて、利用者装置21から証明要求を受付けた証明装置1が、該証明要求に対して受理証明書を含むシーケンス補完方式（登録点に関して、該登録点の即時補完データ及び該登録点より前に登録された各登録点の該登録点における遅延補完データを証明応答に含む）による証明応答を発行し、利用者装置21がこの証明応答を用いて、利用者装置21間における受理証明書発行の時間的前後を検証することができるので、証明要求をまとめた公表データが電子的に公表される前であっても、受理証明書の正当性を検証することができる。

#### 【0085】

<第2の実施の形態>

##### （2-1. システム構成）

図9は、本発明の第2の実施の形態に係るイベント順序証明システム200のシステム構成図である。同図に示すイベント順序証明システム200は、証明装置4、複数の利用者装置51（ $1=A, B, \dots, N$ ）、及び、以上の各装置を相互に接続する、例えば、インターネット網、電話回線網などにより構成されるコンピュータネットワーク3を備えており、証明装置4が利用者装置51からの証明要求に応じて、受理証明書を含む証明応答を利用者装置51に返信するようになっている。そして、利用者装置51は、証明装置4から受け取ったこの複数の証明応答から受理証明書を検証することができるようになっている。

#### 【0086】

ここで、本実施の形態と第1の実施の形態とは、証明応答の形式が異なるものであり、本実施の形態では、「シーケンス補完方式」とは別の後述する「連鎖補完方式」により証明応答が作成されるようになっている。これは、上述したシーケンス補完方式においては、各利用者装置21の各登録点において、当該の利用者装置21の過去の登録点全てに対してその遅延補完データを当該の利用者装置21に返送しなければならないため、過去の登録点が増加するのに比例して、証明応答のデータ量が増加するが、本実施の形態の連鎖補完方式においては、証明応答のデータ量の増加が抑制されるようになっている。尚、本実施の形態においては、上記実施の形態と異なる構成及び機能を説明し、その他の構成及び機能に関しては同一部分には同一符号を付して説明を省略する。

#### 【0087】

証明装置4は、コンピュータネットワーク3を介して利用者装置51とデータの送受信を行う送受信部11、利用者装置51からの証明要求として送信されたデジタル・データを順次集約木を用いてまとめるイベント順序証明要求集約部12、受理証明書を含む証明応答を作成するイベント順序証明応答作成部41、証明装置4が一定期間に発行した複数の受理証明書の内容を集約したデータに対して高強度デジタル署名し、公表データとする高強度デジタル署名作成部14、高強度デジタル署名を付加された公表データを電子的に公表する電子的情報公表部15、及び受理証明書をはじめとするイベント順序証明に関する情報を記憶する記憶部42を有する構成である。

#### 【0088】

イベント順序証明応答作成部41は、図10に示すような受理証明書  $E0C(y)$  を含む証明応答を作成し、利用者装置51に送信するようになっている。同図によれば、受理証明書  $E0C(y)$  は、利用者から送付されたデジタル・データ  $y$ 、上述した順次割当データ計算手順によりデジタル・データ  $y$  から計算された順次割当データ  $z$ 、順次割当データ  $z$  が割当てられた順次集約木を一意に特定できる順次集約木番号、順次割当データ  $z$  が割当てられた順次集約木のリーフを一意に特定できる順次集約木リーフ番号、およびその時点で取得できる順次集約補完データの一部（これを登録点の即時補完データと言う）  $SK$  の位置情報及び割当値を含むように構成されている。

#### 【0089】

また、同図によれば、証明応答には、利用者装置21の直前の登録点の遅延補完データ  $TK$  2の位置情報及び割当値も含むように構成されている。

#### 【0090】

ここで、本実施の形態における証明応答の具体例を図4を用いて説明する。尚、上述したように、この証明応答の形式を「連鎖補完方式」と呼ぶ。今、ある利用者装置51からの登録点が  $X1$ （ノード(0, 2)）、 $X2$ （ノード(0, 11)）、 $X3$ （ノード(0, 18)）、 $X4$ （ノード(0, 21)）、 $X5$ （ノード(0, 29)）、 $X6$ （ノード(0, 31)）であるとする。

#### 【0091】

連鎖補完方式においては各登録点において以下のようなデータが利用者装置51に返されるようになっている。

#### 【0092】

(1)  $X1$ 点における証明応答には、 $X1$ 点の即時補完データが返される（具体的には、ノード(1, 0)の割当値）。

#### 【0093】

(2)  $X2$ 点における証明応答には、 $X2$ 点の即時補完データ、及び $X1$ 点の $X2$ 点における遅延補完データが返される（具体的には、 $X2$ 点の即時補完データとして、ノード(3, 0)、ノード(1, 4)、ノード(0, 10)の割当値、 $X1$ 点の $X2$ 点における遅延補完データとして、ノード(0, 3)、ノード(2, 1)の割当値）。

#### 【0094】

(3)  $X3$ 点における証明応答には、 $X3$ 点の即時補完データ、及び $X2$ 点の $X3$ 点における遅延補完データが返される（具体的には、 $X3$ 点の即時補完データとして、ノード(4, 0)、ノード(1, 8)の割当値、 $X2$ 点の $X3$ 点における遅延補完データとして、ノード(2, 3)の割当値）。

#### 【0095】

(4)  $X4$ 点における証明応答には、 $X4$ 点の即時補完データ、及び $X3$ 点の $X4$ 点における遅延補完データが返される（具体的には、 $X4$ 点の即時補完データとして、ノード(4, 0)、ノード(2, 4)、ノード(0, 20)の割当値、 $X3$ 点の $X4$ 点における遅延補完データとして、ノード(0, 19)の割当値）。

#### 【0096】

以下、 $X5$ 及び  $X6$  においても同様である。このように、連鎖補完方式においては、証明応答として、ある登録点に関して、該登録点の即時補完データ及び該登録点の直前登録点

の該登録点における遅延補完データが含まれるようになっている。これにより、連鎖補完方式においては、過去の登録点が増えても証明要求に対する証明応答のデータ量が比例的に増加しないので、証明装置4と利用者装置5i間の通信データ量を抑制することができる。尚、各証明応答は、利用者装置2iごとに管理されるようになっている。

#### 【0097】

次に、連鎖補完方式の証明応答であっても、利用者装置5iは、実質的には、シーケンス補完方式の証明応答と同一のデータが得られることを図11及び図12を参照して説明する。

#### 【0098】

図11に示すように、順次集約木 ST2 の3つのリーフa1, a2, a3 が左からこの順番であるものとする。a2点のa3点における遅延補完データ、a2点の即時補完データ、及びa1点のa2点における遅延補完データから、a1のa3点における遅延補完データは、以下のよう

#### 【0099】

まず、a2点のa3点における遅延補完データの中で最もレベルが高いノードのレベルをj2と置く。また、a1のa2による認証点をAP(a1, a2)、その兄弟ノードをAP'(a1, a2)と書き、AP(a1, a2)のレベルをj1と置く。

#### 【0100】

このとき、第1に、a1点のa3点における遅延補完データに含まれる認証バスノードのうちレベルがj1よりも小さいものの割当値は、a1点のa2点における遅延補完データに含まれる。

#### 【0101】

また、第2に、a1点のa3点における遅延補完データに含まれる認証バスノードのうちレベルがj1に等しいものの割当値は、a2点のa3点における遅延補完データおよびa2点の即時補完データから計算できる。

#### 【0102】

さらに、第3に、a1点のa3点における遅延補完データに含まれる認証バスノードのうちレベルがj1により大きいものの集合は、a2点のa3点における遅延補完データに含まれる認証バスノードのうちレベルがj1より大きいものの集合と等しい。従って、a1点のa3点における遅延補完データに含まれる認証バスノードのうちレベルがj1より大きいものの割当値は、a2点のa3点における遅延補完データから計算できる。

#### 【0103】

以上から、a1点のa3点における遅延補完データは、以下の3つのデータから計算することができる。

#### 【0104】

- (1) a2点のa3点における遅延補完データ
- (2) a2点の即時補完データ
- (3) a1点のa2点における遅延補完データ

尚、上記の(1), (2), (3) からa1点のa3点における遅延補完データを計算する処理を完全化波及処理と呼ぶ。

#### 【0105】

この完全化波及処理を用いることにより、利用者装置5iは、連鎖補完方式の証明応答からシーケンス補完方式の証明応答を計算できるようになっている。図12はこの計算方法を示す表である。ここで、図12は、シーケンス補完方式及び連鎖補完方式において、各登録点に必要な即時補完データ及び遅延補完データを示している。尚、連鎖補完方式における証明応答の即時補完データ及び遅延補完データは、2重線に囲まれる部分のデータである。また、図12に示す矢印は計算の方向を示している。例えば、図12によれば、a2のa3における遅延補完データ(P1)、a2の即時補完データ(P2)、及びa1のa2における遅延補完データ(P3)から、a1のa3における遅延補完データ(P4)が計算できることを示している。



同様に、 $a_3$ の $a_4$ における遅延補完データ(P5)、 $a_3$ の即時補完データ(P6)、及び $a_2$ の $a_3$ における遅延補完データ(P1)から、 $a_2$ の $a_4$ における遅延補完データ(P7)が計算できる。そして、これを繰り返すことにより、連鎖補完方式であっても、最終的には、図12に示す遅延補完データすべてが計算できることになる、これは、シーケンス補完方式の証明応答に他ならない遅延補完データである。

#### 【0106】

従って、本実施の形態の連鎖補完方式であっても、利用者装置51においては、図12に示すような完全化波及処理を行うことにより、第1の実施の形態と同様の検証をすることができるものである。尚、この完全化波及処理に関しては、後述する「インクリメンタル完全化」の処理として、詳しく説明する。

#### 【0107】

利用者装置51は、コンピュータネットワーク3を介して証明装置4とデータを送受信する送受信部21、所定のデジタル・データを含む証明要求を複数回行うイベント順序証明要求証部22、証明要求に対する証明応答に含まれた受理証明書を検証するイベント順序証明検証部51、受理証明書を含む証明応答をはじめとしてイベント順序証明に関する情報を記憶する記憶部52を有する構成である。

#### 【0108】

ここで、イベント順序証明検証部51は、第1の実施の形態のイベント順序証明検証部23の機能に加えて、後述するインクリメンタル完全化の処理を行う機能を具備するもので、受理証明書に対して以下の検証機能を備える。

#### 【0109】

第1の検証機能としては、証明装置4のデジタル署名作成部14及び電子的情報公表部15を介して公表される公表情報に、受理証明書に含まれる順次割当データがハッシュ関数を介してリンクすることを検証するものである。具体的には、順次集約木のルート値が公表された値と利用者装置51で計算されたルート値が一致するか否かを検証するものである。

#### 【0110】

第2の検証機能としては、公表情報が公開される前であっても、利用者装置51間における受理証明書発行の時間的前後を検証するものである。

#### 【0111】

以下、第2の検証機能を図5を用いて説明する。

#### 【0112】

今、2つの利用者装置2A及び2Bが、それぞれ、連鎖補完方式により、各登録点の証明応答を取得するものとし、図5に示す $a$ 、 $a_1$ 、 $a_2$ 、 $a_1$ を利用者装置2Aの登録点、 $b$ を利用者装置2Bの登録点とする。尚、 $a_1$ は、暫定終端点である。図5においては、利用者装置2Aの1つの登録点 $a$ があり、それより右に位置する利用者装置2Bの登録点 $b$ があり、さらにその右に位置する利用者装置2Aの登録点 $a_1$ がある。

#### 【0113】

本実施の形態においては、まず、利用者装置2Aの登録点 $a$ に対して、 $a_1$ を暫定終端点として、図12に示す完全化波及処理を行う。これにより、シーケンス補完方式と同一の証明応答が得られるので、その後は、第1の実施の形態のときと同一の方法を実施することにより、登録点 $a$ と登録点 $b$ の時間的前後を検証することができる。即ち、本実施の形態においても、登録点 $a_1$ を現時点としたときの現時点順次集約木における、登録点 $a$ の登録点 $b$ による認証点 $o$ の割当値 $V(o)$ が一致すれば、登録点 $a$ の登録が登録点 $b$ の登録より前に起こったことを客観的に証明することができる。

#### 【0114】

尚、以上の各装置は、少なくとも演算機能及び制御機能を備えた中央処理装置(CPU: Central Processing Unit)、プログラムやデータを収納する機能を有するRAM(Random Access Memory)等からなる主記憶装置(メモリ)、ハードディスク(HD)等の電源断時にもデータを記憶し続けることができる2次記憶装置を有する電子的な装置から構成されている。このうち、証明装置4のイベント順序証明応答作成部41及び利用者装置51のイベント

順序証明検証部51の処理は、上記CPUによる演算制御機能を具体的に示したものに他ならない。また、証明装置4の記憶部42及び利用者装置51の記憶部52は、上記主記憶装置あるいは2次記憶装置の機能を備えたものである。

#### 【0115】

##### (2-2. システム動作)

ここで、以上の構成を有するイベント順序証明システム200におけるイベント順序証明方法に関しては、図6における証明装置1及び利用者装置21を、それぞれ証明装置4及び利用者装置51に置き換えたものと同様であるため、説明を省略する。また、イベント順序証明検証方法に関しては、図7及び図8における利用者装置2A及び2Bを、それぞれ利用者装置5A及び5Bに置き換えたものであるとともに、事前ステップとして、利用者装置5A及び5Bそれぞれにおいて、後述するインクリメンタル完全化処理を実行すれば、その後の動作は、図7及び図8の動作と同様であるため、説明を省略する。

#### 【0116】

##### <2-3. 証明装置4のデータ記憶方法>

##### (第1の方法)

次に、連鎖補完方式における証明装置4のデータ記憶方法について、より詳細に説明する。まず、第1の方法は、証明装置4が順次集約木を記憶部42上に構成することにより、上述の連鎖補完の方式を実現する方法（以下、方法Aという。）である。

#### 【0117】

図13は、方法Aを採用したときの記憶部42に記憶されるデータの概略構成を示す図である。図13に示すように、記憶部42には、順次集約木そのもの、即ち、証明要求が割り当てられたノード、計算可能なノードの位置情報及び割当値を記憶するとともに、各利用者装置51ごとに直前登録点の位置情報を記憶するようになっている。

#### 【0118】

この方法Aにおいては、証明要求を証明装置4が受信するたびに、記憶部42上に記憶する順次集約木に、レベルが0のノード、即ちリーフを加え、レベルが1以上のノードで割当値が計算できるものに対しては、そのノードを順次集約木に追加しその割当値を付加して、記憶部42上に順次集約木を構築するようになっている。

#### 【0119】

以下に、方式Aにおける証明装置4の動作を図14を用いて説明する。ここで、図14は、証明装置4のイベント順序証明要求集約部12及びイベント順序証明応答作成部41の機能を示すフローチャート図である。

#### 【0120】

まず、証明装置4は利用者装置51から証明要求を受けると、証明要求から順次割当データを生成し、順次集約木の新しいリーフに割り当てて、新登録点とするとともに、該登録点のノード情報（位置情報及び割当値）を記憶する（ステップS1101、S1103）。

#### 【0121】

次いで、新登録点の即時補完データを即時補完データの定義に従って、記憶部42に記憶された順次集約木から取得する（ステップS1105）。

#### 【0122】

次いで、新登録点の追加により、レベルが1以上のノードで割当値が計算できるものに対しては、割当値を計算して、そのノードの位置情報及び割当値を記憶部41に記憶する（ステップS1107）。

#### 【0123】

次いで、直前登録点の遅延補完データを、遅延補完データの定義に従って、記憶部41に記憶された利用者装置51ごとの直前登録点及び順次集約木から取得する（ステップS1109）。

#### 【0124】

次いで、新登録点を直前登録点とし、該利用者装置51の直前登録点として記憶するとともに、ステップS1105及びS1109で取得した即時補完データ及び遅延補完データを含む証明

応答を作成し、利用者装置51に送信する（ステップS1111, S1113, S1115）。

#### 【0125】

（第2の方法）

次に、第2の方法は、証明装置4が順次集約木を記憶部42上に構成するのではなく、スタック構造を記憶部42に構成することにより、上述の連鎖補完の方式を実現する方法（以下、方法Bという）である。方法Bは、順次集約木の大きさにほぼ比例して必要記憶量が増加する方法Aをさらに改善したものであり、スタック構造を用いて順次集約木のノード割当値、各順序証明要求に対する即時補完データ、及び遅延補完データを計算するので、必要記憶量を減少することができ、証明装置4の記憶部42に収まらない大きさの順次集約木の取り扱いを可能とすることができる。

#### 【0126】

図15は、方法Bを採用したときの記憶部42に記憶されるデータの概略構成を示す図である。図15に示すように、記憶部42には、即時補完データ（位置情報及び割当値）を記憶する第1のスタック421と利用者装置51ごとに遅延補完データを記憶する記憶部422を備えており、利用者装置51ごとに遅延補完データを記憶する記憶部422は、直前登録点（位置情報）423及び遅延補完データ（位置情報及び割当値）を記憶する第2のスタック424で構成されている。尚、第1又は第2のスタックの要素となるようなデータをスタックフレームと呼ぶ。

#### 【0127】

ここで、上記2種類のスタックのうち第1のスタックは、従来から用いられているデータ構造である。例えば、R. Merkle: Secrecy, Authentication, and Public Key Systems, UMI Research Press, 1982. の36ページには二分木のルートノードの割当て値を計算するための再帰的手順  $H(a, b)$  が記載されているが、この再帰的手順を1つのスタックを用いて標準的に実装するとき、このスタックは上記第1のスタックと同様の使われ方をする。

#### 【0128】

以下に、方式Bにおける証明装置4の動作を図16を用いて説明する。ここで、図16は、証明装置4のイベント順序証明要求集約部12及びイベント順序証明応答作成部41の機能を示すフローチャート図である。

#### 【0129】

まず、証明装置4は利用者装置51から証明要求を受けると、証明要求から順次割当データを生成し、順次集約木の新しいリーフに割り当てて、新登録点とする（ステップS1121, S1123）。

#### 【0130】

次いで、新登録点の即時補完データを第1のスタック421から取得する（ステップS1125）。

#### 【0131】

次いで、新登録点の位置情報及び割当値を含むスタックフレームを第1のスタックに追加する（ステップS1127）。その際、新たに第1のスタックに追加されたスタックフレームが、他の利用者装置51において、直前登録点の補完データに対応するものであるときは、該スタックフレームを該当する利用者装置51の第2のスタックに追加する（ステップS1129, S1131）。

#### 【0132】

次いで、第1のスタックに兄弟ノードである2つのノードに対応する2つのスタックフレームが存在する限り、当該の2つのノードの親に当たるノードの位置情報と割当値を含むスタックフレームを生成し、当該の2つのノードに対応するスタックフレームを第1スタックから除き、上記新たに生成したスタックフレームを第1スタックに追加する（ステップS1133, S1135, S1137, S1139）。その際、新たに第1のスタックに追加されたスタックフレームが、他の利用者装置51において、直前登録点の補完データに対応するものであるときは、該スタックフレームを該当する利用者装置51の第2のスタックに追加する（ス

テップS1141, S1143)。

【0133】

次いで、当該の利用者装置51に対する第2のスタックから直前登録点の新登録点における遅延補完データを取得し、当該の利用者装置51に対する第2のスタックを空にする（ステップS1145）。

【0134】

次いで、新登録点を直前登録点とし、該利用者装置51の直前登録点として記憶するとともに、ステップS1125及びS1145で取得した即時補完データ及び遅延補完データを含む証明応答を作成し、利用者装置51に送信する（ステップS1147, S1149, S1151）。

【0135】

上記動作を図4に示す具体例を用いて説明する。尚、新登録点をX4として説明する。

【0136】

まず、証明装置4は利用者装置51から証明要求を受けると、証明要求から順次割当データを生成し、順次集約木の新しいリーフである登録点X4に割り当てて、新登録点とする（ステップS1121, S1123）。

【0137】

次いで、新登録点X4の即時補完データであるノード(3,0)の割当値を第1のスタック421のスタックフレーム0から、ノード(2,4)の割当値を第1のスタック421のスタックフレーム1から、ノード(0,20)の割当値を第1のスタック421のスタックフレーム2からそれぞれ取得する（ステップS1125）。

【0138】

次いで、新登録点(0,21)の位置情報及び割当値を含むスタックフレーム3を第1のスタックに追加する（ステップS1127）。その際、新たに第1のスタックに追加されたスタックフレームが、他の利用者装置51において、直前登録点の補完データに対応するものであるときは、該スタックフレームを該当する利用者装置51の第2のスタックに追加する（ステップS1129, S1131）。

【0139】

次いで、第1のスタックに兄弟ノードに対応する2つのスタックフレームが存在するので（スタックフレーム2及び3）、当該の2つのノードの親に当たるノード(1,10)の位置情報と割当値を含むスタックフレームを新たに生成し、当該兄弟ノードに対応するスタックフレーム（スタックフレーム2及び3）を第1スタックから除き、上記新たに生成したスタックフレームを新たなスタックフレーム2として第1スタックに追加する（ステップS1133, S1135, S1137, S1139）。その際、新たに第1のスタックに追加されたスタックフレームが、他の利用者装置51において、直前登録点の補完データに対応するものであるときは、該スタックフレームを該当する利用者装置51の第2のスタックに追加する（ステップS1141, S1143）。

【0140】

次いで、当該の利用者装置51に対する第2のスタックから直前登録点X3の新登録点X4における遅延補完データであるノード(0,19)の割当値を第2のスタック424のスタックフレーム0から取得し、当該の利用者装置51に対する第2のスタックを空にする（ステップS1145）。

【0141】

次いで、新登録点X4を直前登録点X3とし、該利用者装置51の直前登録点423として記憶するとともに、ステップS1125及びS1145で取得した即時補完データ及び遅延補完データを含む証明応答を作成し、利用者装置51に送信する（ステップS1147, S1149, S1151）。

【0142】

（第2の方法の実装例）

以下に、上述した方法Bを用いた証明装置4の証明応答作成の一実装例を説明する。

【0143】

図17は証明装置4の記憶部42の構成を示す図である。図17に示すように、順次集約木の

ノード割当値を計算するノード割当値計算用スタック（以下では-stackと呼ぶ）及び遅延補完用データ構造の配列（以下では-chain-comple-data-vec と呼ぶ）を持つ。-stackの要素はスタックフレームからなるスタック構造であり、各スタックフレームはplace部とvalue部からなる。このうちplace部は順次集約木のノード位置を表すブレースを保持し、そのノードのレベルを示すlevel部とレベル内の番号を示すindex部の組からなる。value部はそのノードの割当値を保持する。-chain-comple-data-vec はデータ構造chain-comple-dataの配列であり、データ構造chain-comple-data は late-ccomple-stack部、prev-point部、prev-point-old部、及び old-tree-id部からなる。このうち、late-ccomple-stack部は-stackと同様にスタックフレームからなるスタック構造であり、prev-point部は直前の登録点を表す識別番号（非負整数またはnil）を表し、prev-point-old部は現在の集約木より前に生成された集約木における直前の登録点を表す識別番号（非負整数またはnil）を表し、old-tree-id部はprev-point-old部がnilではないときprev-point-old部が示す登録点が属する集約木の識別番号（非負整数または nil ）である。

#### 【0144】

図18は、証明装置4における連鎖補完の手順 GET-REQを示す。この手順で使用される変数及び関数は以下のとおりである。

#### 【0145】

- ・ v0 はデジタル・データ（通常はハッシュ値）を保持する変数である。

#### 【0146】

- ・ idx0 は利用者識別番号を表す整数を保持する変数である。

#### 【0147】

- ・ -lev0-ptr は次に受付けたイベント順序証明要求を割当てするリーフの識別番号を保持する変数であり、初期値は0である。

#### 【0148】

- ・ place0 は順次集約木のノード位置を表すブレースを保持する変数である。

#### 【0149】

- ・ sflm0 はスタックフレームを保持する変数である。

#### 【0150】

- ・ F(v0) はイベント順序証明要求に含まれるデジタル・データv0 を順次集約木リーフに割当てするデータに変換する関数を表す。F(v0) は v0 と等しいとすることにしてもよい、あるいはv0 に所定のハッシュ関数（例えば SHA1）を適用した結果とすることにしてもよい。

#### 【0151】

- ・ -tree-id は順次集約木の識別番号を保持する変数である。

#### 【0152】

- ・ make-stackflm(place0, V0) はブレース placeとデジタル・データ V0を引数とし、place0をplace部として持ち、V0をvalue部としてもつスタックフレームを返す関数である。

#### 【0153】

- ・ stack-bufはある瞬間のスタックの状態を保持する変数である。

#### 【0154】

- ・ handle-chain-comple(id0, tree-id0, idx0, V0, prev-point0, immed-stack-data0, late-stack-data0)

は、順次集約木の識別子 tree-id0、リーフの識別子 idx0、割当てデータ V0、即時補完データ immed-stack-data0、遅延補完データ late-stack-data0 から構成される受理証明書を作成し、利用者識別番号 id0を持つ利用者装置5iに向けて送付する関数を表す。

#### 【0155】

証明装置4が証明要求を受信する度に、要求者から受付けたデジタル・データをv0、要求者の識別番号を id0 としてこの手順が呼び出される（ステップST101～ST117）。

#### 【0156】

図19は、図18の手順 GET-REQから呼び出される、ノード値計算処理の手順 COMP-NODE-

VALS を表す。この手順で使用される変数及び関数は以下のとおりである。

【0157】

- ・ `sflmb` はスタックフレームを保持する変数である。

【0158】

- ・ `placelb` はブレースを保持する変数である。

【0159】

- ・ `idxlb` は整数を保持する変数である。

【0160】

- ・ `levlb` は整数を保持する変数である。

【0161】

- ・ `vallb` はデジタル・データ（通常はハッシュ値）を保持する変数である。

【0162】

- ・ `sflm0b` はスタックフレームを保持する変数である。

【0163】

- ・ `place0b` はブレースを保持する変数である。

【0164】

- ・ `lev0b` は整数を保持する変数である。

【0165】

- ・ `lev-nw` 及び `idx-nw` は 整数を保持する変数である。

【0166】

- ・ `floor(x)` は実数  $x$  を引数として、 $x$  を超えない最大の整数を返す関数である。さらに  $y \neq 0$  に対して、 $\text{floor}(x, y) = \text{floor}(x/y)$  と書く。即ち、 $\text{floor}(x, y)$  は  $x$  を  $y$  で割った際の整商である。

【0167】

- ・ `hash-comb2(val0, val1)` はビット列で表される2つのデジタル・データ `val0` と `val1` を引数とし、`val0` と `val1` を接続し、所定のハッシュ関数 (SHA1等) を適用した結果を返す関数とする。

【0168】

図20は、図18の手順 GET-REQ及び図19の手順 COMP-NODE-VALSから呼び出される、遅延データ設定処理の手順 REGISTER-COMPLE-DATA を表す。この手順で使用される変数及び関数は以下のとおりである。

【0169】

- ・ `place` はブレースを保持する変数であり、`sflm` はスタック・フレームを保持する変数である。

【0170】

- ・ `id` は利用者識別番号を表す整数を保持する変数である。

【0171】

- ・ `N` は登録された利用者の総数を保持する変数である。

【0172】

- ・ `auth-node-pl(prev-point, place)` は 順次リーフの識別番号である `prev-point` とブレース `place` を引数とし、`place` の割当値が `prev-point` の遅延補完データに含めるべきものであれば `true` をさもなければ `false` を返す関数である。`place = (j, i)` としたとき、`auth-node-pl(prev-point, place)` が `true` となるための 必要十分条件は、 $\text{floor}(\text{prev-point}, 2^j)$  が偶数で、 $i = \text{floor}(\text{prev-point}, 2^j) + 1$  となることである。

【0173】

次に、図21及至図23を参照しながら、1つ集約間隔が終わり、該集約間隔に対する順次集約木を終端し、次の集約間隔に対する次の順次集約木を初期化する順次集約木切替処理について説明する。

【0174】

図21は順次集約木切替処理のメインルーチンを表す。このルーチンにおいては、図22で

記述されたサブルーチンTERMINATE-STREE-SUB1と図23で記述されたサブルーチンTERMINATE-STREE-SUB2を実行後、大域変数-tree-idを1だけインクリメントし、2つの大域変数-lev0-ptrと-stackを初期化した後終了する。これらの大域変数は図18で記述した手順GET-REQで用いたものである。

#### 【0175】

図22で記述するTERMINATE-STREE-SUB1は、その集約間隔が終了する時点で構成されている順次集約木を基に必要に応じてノードを追加し、それらのノードには前以て定めた手順に従って所定のハッシュ値を割当、当該の集約木のルート値を定義する処理を記述している。

#### 【0176】

図24を参照しながら、TERMINATE-STREE-SUB1の動作を、特定の状況に則して具体的に説明する。リーフ識別番号9の処理が終了後、リーフ識別番号10の処理に入る前に集約間隔が終了し、TERMINATE-STREE-SUB1を呼び出す状況を考える。開始の時点では-stackの状態は、トップから見て[(1, 4), V(1, 4)]と[(3, 0), V(3, 0)]を含むものとなっている。手順TERMINATE-STREE-SUB1により、ノード(1, 5)と(2, 3)にダミーのハッシュ値を割当て、この順次集約木のルート値を定め、同時に各登録点の完全認証バスデータ(ルート値が計算できる認証バスノードの集まり)を定義することが、以下の(1)~(4)に示すようにできる。

#### 【0177】

(1) ステップST1211により、局所変数 sflm-xb に [(1, 4), V(1, 4)] が設定され、-stack は [(3, 0), V(3, 0)] のみを含むものとなる。次にステップST1212により局所変数 place-xb に (1, 4) が、idx-xb に 4 が設定される。

#### 【0178】

ステップST1213では idx-xb が偶数か否かの判定をするが、現在の idx-xb の値 4 は偶数であるので、ステップST1214に進む。

#### 【0179】

ステップST1214では -stack が nil であるか否か判定するが、-stack は nil ではないのでステップST1215に進む。

#### 【0180】

ステップST1215では、局所変数 lev-xb に 1 を設定し、val-xb に V(1, 4) を設定し、idx-lb に 1+4=5 を設定する。さらに、place-lb に (lev-xb, idx-lb) = (1, 5) を設定。この位置情報 (1, 5) が最初のダミー・ノードを表す。この最初のダミー・ノードに割当てたハッシュ値を計算する関数 dummy-hash を呼びだし、その返り値を dum-val-lb に設定する。さらに sflm-lb に

make-stackflm(place-lb, dum-val-lb) = [(1, 5), dum-val-lb]

を設定する。ここで make-stackflm はノードの位置情報とハッシュ値を引数としてスタックフレームを生成する関数である。ここで place-lb と sflm-lb を引数として REGISTER-COMPLE-DATA (図20で定義したもの) を呼び出す。

#### 【0181】

続いて、ステップST1216において、lev-nw に 2 を設定し、idx-nw に floor(idx-xb, 2) = floor(4, 2) = 2 を設定し、さらに place-nw に (2, 2) を設定する。続いて、val-nw に hash-comb2(val-xb, dum-val-lb) を設定する。hash-comb2 は 2 つのハッシュ値を引数として、それらを接続し、所定のハッシュ関数を適用した結果を返す関数である。sflm-nw に make-stackflm(place-nw, val-nw) を設定し、sflm-nw を -stack にプッシュする。これにより、-stack は [(2, 2), V(2, 2)], [(3, 0), V(3, 0)] を含む構成となる。更に place-nw と sflm-nw を引数として REGISTER-COMPLE-DATA を呼び出す。ここでステップST1211に戻る。

#### 【0182】

(2) ステップST1211において -stack を 1 つポップし、sflm-xb に [(2, 2), V(2, 2)] を設定する。(ここで -stack の状態は、[(3, 0), V(3, 0)] となる。) 更に place-xb に

(2, 2) を、idx-xb に2 を設定する。

#### 【0183】

ステップST1213 において idx-xbの値2は偶数と判定されステップST1214に進み、-stackはnilではないので、さらにステップST1215に進む。lev-xb に2を設定し、val-xb に V(2, 2) を設定し、idx-lb に3 を設定し、place-lb に (2, 3) を設定し、dum-val-lbに dummy-hashの返し値を設定し、この値をV(2, 3)と表す。さらにsflm-lb に

make-stackflm(place-lb, dum-val-lb) = [(2, 3), V(2, 3)]

を設定する。ここでplace-lb と sflm-lb を引数として REGISTER-COMPLE-DATA を呼び出す。

#### 【0184】

続いて、ステップST1216において、lev-nw に3を設定し、idx-nw に floor(val-xb, 2) = floor(2, 2) = 1 を設定し、さらにplace-nw に (3, 1) を設定する。続いて、val-nw に hash-comb2(val-xb, dum-val-lb) を設定し、この値をV(3, 1)と書く。 sflm-nw に make-stackflm((3, 1), V(3, 1))を設定し、sflm-nw を -stack にプッシュする。これにより、-stack は [ (3, 1), V(3, 1) ], [ (3, 0), V(3, 0) ] を含む構成となる。さらに place-nw と sflm-nw を引数としてREGISTER-COMPLE-DATA を呼び出す。ここでステップST1211に戻る。

#### 【0185】

(3) ステップST1211 において -stack を1つポップし、sflm-xb に [ (3, 1), V(3, 1) ] を設定する。(-stackの状態は、[ (3, 0), V(3, 0) ] となっている。) さらに、place-xb に (3, 1) を設定し、idx-xb に1 を設定する。

#### 【0186】

ST1213 において idx-xbの値1は奇数と判定され、ST1217に進み、lev-xb に3を設定し、val-xb にV(3, 1) を設定する。-stack をポップしポップされた [ (3, 0), V(3, 0) ] をsflm-lbに設定する。さらに、place-lb に (3, 0) を、lev-lb に 3 を、idx-lb に0を、val-lb にV(3, 0) を設定する。

#### 【0187】

ステップST1218 において、lev-nw に4 を設定し、idx-nw に floor(1, 2) = 0 を設定し、place-nw に (4, 0)を設定し、val-nw に hash-comb2(V(3, 0), V(3, 1)) を設定する。この値をV(4, 0)とする。 sflm-nw にmake-stackflm((4, 0), V(4, 0))を設定し、sflm-nw を -stack にプッシュする。これにより、-stack は [ (4, 0), V(4, 0) ]を含む構成となる。 さらに place-nw と sflm-nw を引数としてREGISTER-COMPLE-DATA を呼び出す。ここでステップST1211に戻る。

#### 【0188】

(4) ステップST1211において、-stack を1つポップし、sflm-xb に [ (4, 0), V(4, 0) ] を設定する。(-stackの状態は nil となっている。) さらに、place-xb に (4, 0) を設定し、idx-xb に0 を設定する。ステップST1213 においてidx-xbの値0は偶数と判断され、ステップST1214に進み -stack はnilであるので、ステップST1219に進み、返し値に V(4, 0) を設定して終了する。

以上の結果、この手順の返し値は 当該の順次集約木のルート値である V(4, 0)となる。

#### 【0189】

次に、図23で記述されたサブルーチンTERMINATE-STREE-SUB2について説明する。

#### 【0190】

まず以下の変数および定数が使用される。

#### 【0191】

- ・ 利用者装置の識別子である非負整数を保持する変数 id
- ・ 利用者装置の総数を表す定数 N
- ・ 図17の遅延補完用スタックの 配列 -chain-comple-data-vec の各要素と同じ構造を持つchain-comple-data2
- ・ 非負整数またはnilを保持する変数prev-chain-point2。



【0192】

次にTERMINATE-STREE-SUB2の動作について説明する。

【0193】

$id = 0, \dots, N-1$  について図23のブロック1を実行する。

【0194】

ブロック1の中では以下の動作をする。

【0195】

chain-comple-data2 に -chain-comple-data-vec[id] を設定し、prev-chain-point2 にchain-comple-data2のprev-point部を設定する（ステップST1222）。

【0196】

prev-chain-point2がnil であればブロックを終了する（ステップST1223）。

【0197】

さもないければ、chain-comple-data2 のprev-point-old部をprev-chain-point2 に、prev-point部をnil に、old-tree部に現在の集約木識別子を各々設定し、次に-chain-comple-data-vec[id] にchain-comple-data2 を設定する（ステップST1224）。

【0198】

<2-4. 利用者装置51におけるインクリメンタル完全化>

次に、利用者装置51におけるインクリメンタル完全化の処理について、詳しく説明する。ここで、インクリメンタル完全化には、大別して(1)インクリメンタル個別完全化と(2)インクリメンタル一括完全化の2つの処理があり、いずれかが利用者装置51において実行されるものである。尚、上述した「完全化波及処理」は、インクリメンタル一括完全化の中の一機能を説明したものである。

【0199】

(インクリメンタル個別完全化)

利用者装置2Aのある集約間隔 $l$ に属するか或いは或いはその次の集約間隔の最初の登録点であるよう登録点 $a_l$ を暫定終端点とする。

【0200】

$a_l$  がある集約間隔 $l$ の次の集約間隔の最初の登録点であるとき、当該の集約間隔 $l$ に対する追伸点と呼ぶ。

【0201】

集約間隔 $l$ の間に、この暫定終端点 $a_l$ までに登録した登録点の集合を

$a(0), a(1), \dots, a(n)$  とする。

【0202】

(通常は、 $a(n) = a_l$ であるが、 $a_l$  が追伸点であるときはこうはならない。)

このとき、登録点の集合から、 $a_l$ を暫定終端点としたときの1つあるいは複数の順次集約小木が構成され、構成された順次集約小木の集まりを、 $a_l$ を暫定終端点としたときの順次集約フォレストと呼ぶことにする。

【0203】

図25を参照して、順次集約フォレストと順次集約小木について具体的に説明する。

【0204】

$a_l$  が当該の集約間隔にあるとき、 $a_l$  がリーフ番号（非負整数）を表し、 $a_l$  の二進表現においては $k(1), k(2), \dots, k(m)$  桁において1が立つものとする（但し最小桁は0桁とする）。図25においては、 $m = 4$  としている。ここで、 $k(m)$  は0でもよく、 $k(1) > k(2) > k(3) > \dots > k(m)$  とならなければならない。このとき、上記順次集約フォレストに属する $n$ 番目の順次集約小木のリーフの数は  $2^{k(n)}$  となる。図25においては、ST2(1), ..., ST2(4) が順次集約小木を表している。

【0205】

インクリメンタル個別完全化とは、指定された $a \in \{a(0), a(1), \dots, a(n)\}$  について、以下の計算(1)、(2)、(3)を行うことである。

【0206】

(1)  $a$  が属する順次集約小木  $ST$  (ユニークに定まる) を計算する。

【0207】

(2) 順次集約フォレストにおいて上記  $ST$  より左に位置する 1 つあるいは複数の順次集約小木のルートの割当値を、 $a(0), \dots, a(n)$  (及び  $a_f$ ) で取得する補完データから計算する。

【0208】

(3)  $ST$  における  $a$  の認証パスを  $authPath_{ST}(a)$  とおくと、 $authPath_{ST}(a)$  に属するノードの割当値を、 $a(0), \dots, a(n)$  (及び  $a_f$ ) で取得する補完データから計算する。

【0209】

尚、インクリメンタル個別完全化の定義は、以下に定義する現時点順次集約木を用いて行うこともできる。ここで、現時点順次集約木とは、 $a_f$  を暫定終端点としたときの順次集約フォレストを含むような最小の二分木を、 $a_f$  を暫定終端点としたときの現時点順次集約木という。

【0210】

図26は、現時点順次集約木の枝を実線及び点線で示すものである。このうち、点線で示す枝が順次集約フォレストには含まれないが現時点順次集約木を構成するために追加した枝である。また黒で塗りつぶされた小円が順次集約フォレストに含まれるノードであり、塗りつぶしがない小円が現時点順次集約木を構成するために追加したノードである。現時点順次集約木を  $CST$  と表し、 $CST$  における  $a$  の認証パスを  $authPath_{CST}(a)$  とおく。このとき、インクリメンタル個別完全化とは、指定された  $a \in \{a(0), a(1), \dots, a(n)\}$  について、 $authPath_{CST}(a)$  に属し、既に割当値が定まっているノードの割当値を、 $a(0), \dots, a(n)$  (及び  $a_f$ ) で取得する補完データから計算することと同値である。

【0211】

尚、順次集約フォレストのリーフの数の総数を  $N$  としたとき、現時点順次集約木の高さ  $h$  は  $N \leq 2^k$  となる最小の非負整数  $k$  となる。

【0212】

上述したことを踏まえて、第2の検証機能を説明した図5に戻ると、登録点  $a$  に対して、 $a_f$  を暫定終端点としてインクリメンタル個別完全化を行うと、認証点  $o$  の割当値  $V(o)$  が計算できることになる。これにより、登録点  $a_f$  を現時点としたときの現時点順次集約木における、登録点  $a$  の登録点  $b$  による認証点  $o$  の計算された割当値  $V(o)$  が、登録点  $b$  で取得される即時補完データに含まれれば、登録点  $a$  の登録が登録点  $b$  の登録より前に起こったことを客観的に証明することができる。

【0213】

次に、図27及び図28を参照しながら、インクリメンタル完全個別化の動作について説明する。図27は、インクリメンタル完全個別化の動作を示すフローチャート図である。

【0214】

まず、暫定終端点  $a_f$  より以前の登録点で集約間隔  $l$  に属する登録点  $a$  を 1 つ指示する (ステップ S510)。図28に示す順次集約フォレストにおいては、インデックス18のリーフ (ノード (0,18)) を  $a$  としている。

【0215】

次に、登録点  $a$  が属する順次集約小木  $ST$  を計算する (ステップ S520)。図28に示す順次集約フォレストにおいては、構成済み順次集約小木のうち左から2番目の  $ST2(2)$  が  $ST$  となる。

【0216】

次に、登録点  $a$  の  $ST$  における各遅延認証パスノード  $s$  を計算する (ステップ S530)。図28に示す順次集約フォレストにおいては、インデックス19のリーフ (ノード (0,19)) とレベル2のインデックス5のノード (ノード (2,5)) が遅延認証パスノードとなる。

【0217】

次に、各遅延認証パスノード  $s$  の取得参照点をそれぞれ決定する (ステップ S540)。ここで、取得参照点及び取得タイミング点について説明する。尚、以下では、ある登録点  $a$

の即時補完データと、 $a$  の次の登録点  $a_1$  で取得される  $a$  の遅延補完データの合併を、登録点  $a$  の連鎖補完データとよぶ。

#### 【0218】

登録点  $a_0$  が与えられたとき、その  $authPath(a_0)$  でレベル  $j$  のノード  $(j, s(j))$  の割当値  $V(j, s(j))$  を計算するために十分な情報を補完データから直接取得できるかあるいは該データを基に計算により取得できる要求登録点を  $V(j, s(j))$  の取得参照点と呼ぶ。そして、上記の必要な補完データを受信する登録点を取得タイミング点と呼ぶ。

#### 【0219】

例えば、図4に示す順次集約木において、登録点  $X_3$  が与えられたとき、遅延補完データの1つであるノード  $(0, 19)$  の取得参照点は、登録点  $X_3$  (ノード  $(0, 18)$ )、取得タイミング点は、登録点  $X_4$  (ノード  $(0, 21)$ ) となる。また、登録点  $X_3$  が与えられたとき、即時補完データの1つであるノード  $(4, 0)$  の取得参照点及び取得タイミング点は、登録点  $X_3$  (ノード  $(0, 18)$ ) となる。一般に、即時補完データに関しては、取得参照点と取得タイミング点は一致する。

#### 【0220】

次に、ステップS540で決定された取得参照点をもとに、各認証パスノード  $s$  の割当値を計算する(ステップS550)。

#### 【0221】

以上により、登録点  $a$  におけるインクリメンタル完全化の処理が終了し、この計算結果を用いて、登録点  $a$  の割当値を含む入力に衝突困難ハッシュ関数を適用し、 $V(\text{root}(ST))$  を計算することができる。尚、上述した計算が可能であるのは、 $s$  の割当値の取得参照点および取得タイミング点がともに、暫定登録点  $a_1$  以前であることによる。

#### 【0222】

(インクリメンタル個別完全化の実装例)

以下に、上述したインクリメンタル個別完全化の一実装例を説明する。

#### 【0223】

まず、順次集約小木  $ST$  を決定する計算手順 FOREST-SST を図29に示す。これは、図27のステップS520に相当するものである。

#### 【0224】

この手順で使用される変数及び関数は以下のとおりである。

#### 【0225】

- ・入力としては、リーフ識別子  $a$  (非負整数)、暫定終端点の識別子  $fin$  (非負整数)
- ・出力としては、 $a$  が属する順次集約小木の一番左のリーフ識別子  $start$  (非負整数)、 $a$  が属する順次集約小木の一番右のリーフ識別子  $last$  (非負整数)
- ・変数としては、非負整数を保持する変数  $rest, ht, leaf-num$
- ・使用される関数として、 $\log_2(x)$  は、 $\log_2(x)$  以下の最大の整数、 $\text{expt}(x, y)$  は、 $x$  の  $y$  乗

このアルゴリズムは、リーフ識別子  $a$  (非負整数) と暫定終端点の識別子  $fin$  (非負整数) を入力とし、該暫定終端点の登録が終了した時点における順次集約フォレストに属する順次集約小木で  $a$  が属するものを  $ST$  としたとき、 $ST$  の一番左に位置するリーフの識別子  $start$  (非負整数) と一番右に位置するリーフの識別子  $last$  (非負整数) の組を出力するものである。当該の順次集約小木  $ST$  のリーフの数は、 $last - start + 1$  となり、当該の順次集約小木の高さは、 $\log_2(last - start + 1)$  となる。

#### 【0226】

図28で示した具体例について、図29の手順を適用すると以下ようになる。リーフ識別子  $a$  を18とおき、暫定終端点の識別子  $fin$  を26とおく。このとき、図29の手順に従って計算すると、 $start$  として16が返され、 $last$  として23が返される。この出力から、 $a$  が属する順次集約小木は、図28の  $ST_2(2)$  であることが分かる。

#### 【0227】

次に、インクリメンタル完全化における認証パスノード割当値の取得参照点決定手順 D

ECIDE-GET-POINT-Aを図30に示す。この手順は、指定された要求登録点の指定されたレベルの認証パスノードの割当値が、どの要求登録点の即時補完データあるいは遅延補完データから得られるかを決定するものであり、図27のステップS540に相当するものである。

【0228】

図31は、手順DECIDE-GET-POINT-Aにおいて使われるデータ構造と変数の一部を表す。データ構造chaindataはleaf-index部、rgt-value部、immediate部、late部からなる構造である。Mを一つの利用者装置51が一つの集約間隔に送出する順序証明要求の最大数とする。

【0229】

この手順で使用されるその他の変数及び関数は以下のとおりである。

【0230】

・変数chaindata-storeは要素数がMの配列で、配列の各要素がデータ構造chaindataを保持するようなものである(図31参照)。

【0231】

・chaindata0はデータ構造chaindataを保持する変数である。

【0232】

・a0は順次集約木のリーフ識別子を表す整数を保持する変数である。

【0233】

・順次集約木のノード(j, i)に対して、subTree(j, i)は(j, i)をルートとするような順次集約木の部分木を表す。

【0234】

・順次集約木の部分木STに対して、leafs(ST)はSTのリーフの集合を表す。例えば、leafs(subTree(j, i))は部分木subTree(j, i)のリーフの集合である。また、height(ST)はSTの高さを表す。

【0235】

次に、図30に示すインクリメンタル完全化の認証パスノード割当値の取得参照点決定手順(DECIDE-GET-POINT-A)のアルゴリズムの根拠を図32に示す図をもとに説明する。

【0236】

ここで、 $a_l$ を暫定終端点とする構成済みフォレスト内の順次集約小木のうちa0が属するものをSTとおく( $a_0 \in \text{leafs}(ST)$ )。

【0237】

また、ST内のa0の認証パスをauthPath<sub>ST</sub>(a0)とおき、

$$\text{authPath}_{ST}(a_0) = [(0, s(0)), (1, s(1)), \dots, (k-1, s(k-1))]$$

と置く。(但し、kは構成済みフォレスト内のaが属する順次集約小木STの高さである。即ち、 $k = \text{height}(ST)$ 。)以下では、非負整数n, mに対して、 $[n \dots m]$ はn以上でm以下の整数の集合を表すものとする。

【0238】

各  $j \in [0 \dots k-1]$  に対して、(j, s(j))がどの登録点の補完データから計算できるかを決定するアルゴリズムを以下に示す。

【0239】

rtPath<sub>ST</sub>(a0)をSTにおけるaのルート・パスとおき、

$$\text{rtPath}_{ST}(a_0) = [(0, r(0)), (1, r(1)), \dots, (k-1, r(k-1)), (k, r(k))]$$

とする。 $r(0) = a_0$ 、 $\text{root}(ST) = (k, r(k))$ である。 $j \in [0 \dots k-1]$ とする。

【0240】

(1) (j, r(j))が(j+1, r(j+1))のleft-childであるとき

$s(j) = r(j) + 1$ であり、(j, s(j))は(j+1, r(j+1))のright-childである。leafs(subTree(j, r(j)))に属する要求登録点の中で、最も右に位置する点を計算し、a1と置く。

【0241】

(1-1)  $a_l \neq a_l$  のとき

$a_l$  の次の要求登録点を  $a_2$  と置く ( $a_l \neq a_f$  ではないのでこのような  $a_2$  が在る。  $a_2 \leq a_f$ )。

【0242】

(1-1-1)  $a_2 \in \text{leaves}(\text{subTree}(j, s(j)))$  のとき

$\text{leaves}(\text{subTree}(j, s(j)))$  に属する要求登録点で最も右に位置するものを  $a_3$  と置く。

【0243】

(1-1-1-1)  $a_3 = \text{last}(\text{leaves}(\text{subTree}(j, s(j))))$  であるとき (図32(a)を参照)

即時補完データ  $\text{immedData}(a_3)$  は、 $\text{subTree}(j, s(j))$  における  $a_3$  の完全補完データ  $\text{completeData}(\text{subTree}(j, s(j)), a_3)$  を含む。従って、

$\text{immedData}(a_3) \vdash V(j, s(j))$

(尚、 $X \vdash Y$  は、 $X$  から  $Y$  が計算できることを表す。)

このとき、 $V(j, s(j))$  の取得参照点および取得タイミング点は共に  $a_3$  としてよい。

【0244】

(1-1-1-2)  $a_3 \neq \text{last}(\text{leaves}(\text{subTree}(j, s(j))))$  であるとき

(1-1-1-2-1)  $a_3 \neq a_f$  であるとき (図32(b)を参照)

$a_3$  の次の要求登録点を  $a_4$  と置く。 ( $a_3 \neq a_f$  なので、このような  $a_4$  がある。  $a_4 \leq a_f$ 。  $a_4$  は追伸点である可能性もある。)

$a_3$  の  $a_4$  による認証点のレベルを  $j'$  とすると、 $j' \geq j+1$

従って、順次集約木の性質により、 $V(j, s(j))$  は  $a_3$  で取得する即時補完データと、 $a_4$  で取得する ( $a_3$  に対する) 遅延補完データ から計算できる。即ち、

$\text{immedData}(a_3) \cup \text{lateData}(a_3, a_4) \vdash V(j, s(j))$

(尚、 $a$  を  $a_3$  の  $a_4$  による認証点とすると、 $\text{immedData}(a_4)$  には  $V(a)$  が入っているが、これが  $V(j, s(j))$  とは限らないことに注意する必要がある。)

このとき、 $V(j, s(j))$  の取得参照点を  $a_3$ 、取得タイミング点を  $a_4$  としてよい。  $a_3 \leq a_f$  かつ  $a_4 \leq a_f$  である。

【0245】

(1-1-1-2-2)  $a_3 = a_f$  であるとき (図32(c)を参照)

このとき、 $a_f$  を暫定終端点とする構成済みフォレストにおいて  $(j, s(j))$  を含む順次集約小木はない。

【0246】

よって、 $ST$  は  $(j, s(j))$  を含むことは無い。従ってこのような場合はあり得ない。

【0247】

(1-1-2)  $a_2 \in \text{leaves}(\text{subTree}(j, s(j)))$  ではないとき (図32(d)を参照)

$V(j, s(j)) \in \text{lateData}(a_l, a_2)$  である。

【0248】

$V(j, s(j))$  の取得参照点を  $a_l$ 、 $V(j, s(j))$  の取得タイミング点を  $a_2$  としてよい。

【0249】

$a_l \leq a_f$  かつ  $a_2 \leq a_f$

(1-2)  $a_l = a_f$  であるとき (図32(e))。

【0250】

このとき、 $a_f$  を暫定終端点とする構成済みフォレスト内に  $(j, s(j))$  を含む順次集約小木はない。

【0251】

よって、 $ST$  は  $(j, s(j))$  を含むことは無い。従って、このような場合はあり得ない。

【0252】

(2)  $(j, r(j))$  が  $(j+1, r(j+1))$  の right-child であるとき (図32(f))

$r(j) = s(j) + 1$  であり、 $(j, s(j))$  は  $(j+1, r(j+1))$  の left-child である。

【0253】

$V(j, s(j)) \in \text{immedData}(a)$

$V(j, s(j))$  の取得参照点および取得タイミング点を共に  $a_0$  としてよい。  $a_0 \leq a_f$

以上から、図30に示すインクリメンタル完全化の認証バスノード割当値の取得参照点決定手順 (DECIDE-GET-POINT-A) は、すべての場合分けを考慮した手順となっているので、図30に示すアルゴリズムは正しいことがわかる。

#### 【0254】

次に、要求登録点 $a$ が与えられたとき、その $\text{authPath}(a)$ に含まれるレベル $j$ のノード( $j, a(j)$ )の割り当て値 $V(j, a(j))$ を計算する手順 COMPLETION-SUB1を図33に示す(但し、 $0 \leq j < k$ で  $k$  は順次集約木の高さ)。これは、図27のステップS550を中心に説明するフローチャートである。

#### 【0255】

この手順で使用される変数及び関数は以下のとおりである。

#### 【0256】

・  $\text{chaindata0}$  及び  $\text{chaindata1}$  はデータ構造  $\text{chaindata}$  を保持する変数である。

#### 【0257】

・  $\text{immedData1}$  及び  $\text{lateData1}$  はデータ構造  $\text{stackfilm}$  の線形リストを保持する変数である。

#### 【0258】

・  $\text{chaindata-store}$  は当該の集約期間における、各登録点において証明応答として受信したデータを格納する配列である。配列の各要素は、図31で定義した  $\text{chaindata}$  の構造を持つ。この配列の  $i$  番目の要素は、当該の集約間隔における  $i$  番目の登録点の即時補完データと、その直後の登録点で取得した  $i$  番目の登録点の遅延補完データを含む。

#### 【0259】

(1) 本手順は2つの引数(入力)をもち、第1の引数を配列  $\text{chaindata-store}$  の  $\text{index}$  を表す整数  $i0$  とし、第2の引数を順次集約木のレベルを表す整数  $j$  とする(ステップS T5501)。

#### 【0260】

(2) 局所変数  $\text{chaindata0}$  に  $\text{chaindata-store}[i0]$  を設定し(ステップST5502)、局所変数  $a0$  を  $\text{chaindata1}$  の  $\text{leaf-index}$  部とし(ステップST5503)、変数  $a1$  に  $\text{authPath}(a0)$  のレベル  $j$  のノードの  $\text{index}$  を設定する(ステップST5504)。

#### 【0261】

(3) 図30に記述の認証バスノード割当値の取得参照点決定手順 DECIDE-POINT-Aにより  $V(j, a(j))$  の取得参照点  $a2$  (要求登録点の1つで、その点で得られた連鎖補完データから、 $V(j, a(j))$  が計算できるもの)を決定する(ステップST5505)。

#### 【0262】

(4) 配列  $\text{chaindata-store}$  を探索し、 $\text{leaf-index}$ 部が  $a2$  であるような配列要素の  $\text{index}$  となる整数  $i1$  を決定する(ステップST5506)。

#### 【0263】

(5) 変数  $\text{chaindata1}$  に  $\text{chaindata-store}[i1]$  を設定する(ステップST5507)。

#### 【0264】

(6) 変数  $\text{rgt-vall}$  に  $\text{chaindata1}$  の  $\text{rgt-val}$ 部を、変数  $\text{immedData1}$  に  $\text{chaindata1}$  の  $\text{immediate}$ 部を、変数  $\text{lateData1}$  に  $\text{chaindata1}$  の  $\text{late}$ 部を、各々設定する(ステップST5508)。

(7)  $\text{rgt-vall}$ 、 $\text{immedData1}$ 、あるいは  $\text{lateData1}$  に  $\text{place}$ 部が  $(j, a1)$  となるスタックフレームが含まれるかどうか判定する(ST5509)。

#### 【0265】

(7-1) 含まれれば、その値を返す(ステップST5510)。

#### 【0266】

(7-2) 含まれないときは、 $\text{immedData1}$ と  $\text{lateData1}$  から、ハッシュ関数を介して計算できるノードの割当値をレベル0からレベル  $j$  まで順次計算する(ステップST5511)。

#### 【0267】

(7-2-1) 上記で順次計算された割り当て値に  $V(j, a(j))$  が含まれるか否か判定する(ス

テップST5512)。

【0268】

(7-2-1-1) 含まれれば、その値を返す (ステップST5513)。

【0269】

(7-2-1-2) 含まなければ、エラーとする (ステップST5514)。

【0270】

図34は、要求登録点 $a$ が与えられたとき、その $\text{authPath}(a)$ に含まれるレベル $j$ のノード $(j, a(j))$ の割り当て値 $V(j, a(j))$ のリスト

$[V(0, a(0)), V(1, a(1)), \dots, V(k-1, a(k-1))]$

を計算する手順 COMPLETION-SUB1を表す(但し、 $0 \leq j < k$ で  $k$  は順次集約木の高さ)

この手順で使用される変数及び関数は以下のとおりである。

【0271】

・  $k$  を順次集約木の高さとする。

【0272】

・  $\text{auth-node-vals}$  は長さ $k$ の配列で、各配列要素はハッシュ値を保持するものとする。

【0273】

まず、図33のCOMPLETION-SUB1を各 $j$  ( $0 \leq j < k$ )に適用し、 $\text{authPath}(a)$ に属する各ノードの割当値を計算し、計算結果を  $\text{auth-node-vals}[j]$  に格納する (ステップST5523, ST5524)。

【0274】

次に、 $\text{auth-node-vals}$  を返り値とし、終了する (ステップST5525)。

【0275】

(インクリメンタル一括完全化)

上述したインクリメンタル完全化の方法は、完全化の対象となる受理証明書を指定し、その指定された受理証明書の個別完全化を行う方法である。次に述べるインクリメンタル完全化の方法は、ある利用者装置51が連続して取得した一連の受理証明書を一括して、上記のインクリメンタル個別完全化により計算されるものと同じデータを計算する方法である。この種類のインクリメンタル完全化をインクリメンタル一括完全化と呼ぶ。即ち、一連の登録点 $a(0), a(1), \dots, a(n)$ 全てに対して上記のインクリメンタル個別完全化により計算されるものと同じデータを一括して計算することを、インクリメンタル一括完全化という。

【0276】

インクリメンタル一括完全化は、上述した完全化波及処理を用いた以下の手順により実行できる。

【0277】

(1)  $a(0), \dots, a(n)$  を、ある集約間隔 $l$ に属する、ある利用者装置51による一連の登録点とする。

【0278】

(2) 利用者装置51による $a(n)$ の次の登録点を  $a_f$  とする。 $a_f$  は追伸点であってもよい。

【0279】

(3) 図35で記述した手順COMPLETION-BULK-BACKWARD1により、各 $a = a(n), \dots, a(0)$  を、この順にインクリメンタル完全化を行う。

【0280】

この手順により、各登録点  $a(n-i)$  (但し  $i=0, \dots, n$ ) に対して $a_f$ を暫定終端点とした場合のインクリメンタル個別完全化が実現されることが、以下のように数学的帰納法により示される (図36及び図37を参照)。以下では、簡単のため $a(0), \dots, a(n)$ が共通の順次集約小木ST2に属し、 $a_f$ はST2の各リーフよりも右に位置する場合を考える。尚、一般の場合も同様である。

【0281】

$i=0, \dots, n$  について、手順COMPLETION-BULK-BACKWARD1により追加されたものを含む $a$

( $n-i$ ) の遅延補完データと、 $a(n-i)$  で受信した即時補完データの合併は、 $a(n-i)$  の ST2 における全ての認証バスノードの割当値を含んでいることを示せばよい。

#### 【0282】

##### (1) 帰納法のベース

$i=0$  のときを考える。このとき、 $a(n-i) = a(n)$  である。手順 COMPLETION-BULK-BACKWARD1 により、 $a(n)$  に対しては  $a_i$  の登録処理の終了時点における  $a(n)$  の遅延遅延補完データが追加される (図35のステップ ST5003)。ここで、 $a_i$  の登録処理の終了時点においては順次集約小木のルート値は確定しているので、手順 COMPLETION-BULK-BACKWARD1 により追加されたものを含む  $a(n)$  の遅延補完データと、 $a(n)$  で受信した即時補完データの合併は、 $a(n)$  の ST2 における全ての認証バスノードの割当値を含んでいる。

#### 【0283】

##### (2) 帰納ステップ

$i! \in \{0, \dots, n-1\}$  とし、 $i=i!$  としたとき、手順 COMPLETION-BULK-BACKWARD1 により追加されたものを含む  $a(n-i)$  の遅延補完データと、 $a(n-i)$  で受信した即時補完データの合併は、 $a(n-i)$  の ST2 における全ての認証バスノードの割当値を含んでいるものと仮定する。 $i=i!+1$  に対しても同じことが成立つことを示せばよい。このことは、以下のよう完全化波及処理を用いることにより示すことができる。

#### 【0284】

$a2 = a(n-i!), a1 = a(n-(i!+1))$  とおき、 $a1$  の  $a2$  による認証点を  $AP(a1, a2)$ 、その兄弟ノードを  $AP'(a1, a2)$ 、さらに  $AP(a1, a2)$  のレベルを  $j!$  と置く (図37を参照)。

#### 【0285】

以下で述べる順次集約木の性質により、順序集約小木 ST2 における  $a1$  の ST2 における認証バスノードのうちレベルが  $j!$  より小さいものの割当値は、図35のステップ ST5004 において追加されるデータに含まれる。

#### 【0286】

$a1$  の ST2 における認証バスノードのうちレベルが  $j!$  に等しいものの割当値は、図35のステップ ST5007 において追加されるデータに含まれる。

#### 【0287】

$a1$  の ST2 における認証バスノードのうちレベルが  $j!$  より大きいものの割当値は、図35のステップ ST5008 において追加されるデータに含まれる。

#### 【0288】

以上により、手順 COMPLETION-BULK-BACKWARD1 により追加されたものを含む  $a1 = a(n-(i+1))$  の遅延補完データと  $a(n-(i+1))$  で受信した即時補完データの合併は、 $a(n-(i+1))$  の ST2 における全ての認証バスノードの割当値を含むことが導かれる。

#### 【0289】

以上 (1) 及び (2) により、 $i=0, \dots, n$  について、手順 COMPLETION-BULK-BACKWARD1 により追加されたものを含む  $a(n-i)$  の遅延補完データと、 $a(n-i)$  で受信した即時補完データの合併は、 $a(n-i)$  の ST2 における全ての認証バスノードの割当値を含んでいることが示される。

#### 【0290】

なお、同様の帰納法により、図35のステップ ST5006 で判定結果が NO となり、エラーとなることはないことも示される。

#### 【0291】

##### (メモリの効率化)

上記は、1つの集約間隔において連鎖補完方式により利用者装置 51 がイベント順序証明応答として取得したデータを計算機のメモリに読み込むことができる場合の処理方式である。集約間隔における利用者装置 51 による登録点が多数に上り上記の取得データを計算機のメモリに読み込むことができない場合には、以下の方式により取得データの一部をメモリに読み込み、完全認証バスデータを段階的に計算することにより、当該の集約間隔における全ての登録点の完全認証バスデータを計算することができる。



#### 【0292】

上記の計算は、以下のステップ(1)及至(5)により行われる。

#### 【0293】

(1) ある集約間隔にある利用者装置51が受信した補完データのうちから、登録点のインデックスが特定の条件を満たすもののみを間引き抽出し、間引き抽出データを構成する。

#### 【0294】

間引き抽出するための条件として、間引き間隔となる正整数 $m$ を指定し、登録点のインデックスが $m$ で割り切れるもののみを抽出することにしてもよい。図38に示す具体例においては、インデックス0から10までをもつ登録点(黒い点で示されている)から、間引き間隔を5として、5で割り切れるインデックス即ち0, 5, 10を持つ登録点を抽出し、間引き抽出データを構成している。

#### 【0295】

(2) 登録点のインデックスが隣あう間引き抽出データのインデックスで挟まれているような登録点の登録値および補完データからなる局所データを構成する。このような局所データは、一般には複数構成される。

#### 【0296】

図39に示す具体例においては、間引き抽出データの1番目のインデックス0と2番目のインデックス5に挟まれたインデックスを持つ登録点を集め第1の局所データを構成している。図40に示す具体例においては、間引き抽出データの2番目のインデックス5と3番目のインデックス10に挟まれたインデックスを持つ登録点を集め第2の局所データを構成している。

#### 【0297】

(3) 上記(2)で構成された各局所データを、その局所データの最も右に位置する登録点を暫定終端点として扱い、前記のインクリメンタル完全化の処理を行う。この処理を局所データの局所完全化と呼ぶ。

#### 【0298】

この処理により計算される遅延補完データを用いて、当該の局所データに属する登録点のうち最も右に位置するものを $a1$ としたとき、当該の局所データに属する各登録点 $a$ について、 $a$ の $a1$ による認証点を $AP(a, a1)$ としたとき、 $a$ の認証パスノードの中で、 $level(AP(a, a1))$ より低いレベルのものに対しては、その割当値を計算することができる。さらに、ここで割当値を計算できる $a$ の認証パスノードの割当値は、 $a$ の認証パスノードのうち $a1$ の処理が終了した時点で割当値が定まっているもの全てを含んでいる。即ち、登録点 $a$ の $a1$ 時点の遅延補完データを含んでいる。

#### 【0299】

図39に示す具体例においては、この局所データの局所完全化の処理により、インデックス0の登録点の遅延補完データとして、ノード(1, 0)、(2, 1)の割当値が計算され、インデックス1の登録点の遅延補完データとして、ノード(2, 1)の割当値が計算され、インデックス3の登録点の遅延補完データとして、ノード(1, 5)の割当値が計算される。

#### 【0300】

この局所データに属する登録点のうち最も右に位置するものはインデックス5の登録点でありこれを $a1$ と置くと、この局所データに属する各登録点 $a$ について、 $level(AP(a, a1))$ より小さいレベルの認証パスノードの割当値が計算できる。例えば、 $a$ をインデックス0の登録点とすると、 $AP(a, a1)$ は(3, 0)であり、 $a$ の認証パスノードのうちレベルが3より小さいもの(0, 0)、(1, 0)、(2, 1)の割当値を計算できる。 $a$ の $a1$ における遅延補完データは(1, 0)、(2, 1)の割当値であるので、 $a1$ 時点における遅延補完データは計算できることがわかる。この局所データに属する他の登録点についても同様である。

(4) 上記(3)の処理の結果、間引き抽出データの隣あう2つの登録点 $a1$ と $a2$ について、 $a1$ の $a2$ 時点における遅延補完データを取得することが出来る。このデータを用いて前記の証明書完全化のメインルーチン COMPLETION-MAIN1 を適用し、間引き抽出データに含まれる各登録点で取得する証明書の完全化、即ち該登録点の全ての認証パスノードの割当値

を計算する。この処理を間引き抽出データの大域完全化と呼ぶ。

#### 【0301】

図41で示す具体例においては、間引き抽出データのインデックス0, 1, 2の登録点、即ち、リーフ番号1, 11, 31の登録点の証明書の完全化を行い、当該の3つの登録点の全ての認証パスノードの割当値を計算することができる。例えば、間引き抽出データのインデックス0の登録点の認証パスノードは(0, 0), (1, 1), (2, 1), (3, 1), (4, 1)であるが、これら全ての割当値を計算することができる。

#### 【0302】

(5) 上記(3)で構成した局所完全化された局所データの各々と、上記(4)で構成した大域完全化された間引き抽出データを用いて、当該の局所データに含まれる各登録点で取得する証明書の完全化を行う。この処理を局所データの大域完全化と呼ぶ。

#### 【0303】

局所データの大域完全化の手順は、詳しくは以下の通りである。

#### 【0304】

(5-1) ある局所データの登録点を  $a(0), a(1), \dots, a(n) = a_l$  とするとき、 $a_l$  の全ての認証パスノードの割当値は、ステップ(4)で計算済みである。これを、

$V(0), V(1), \dots, V(k-1)$  とする。

#### 【0305】

(5-2) 従って、 $a_l$  のルートパスに属する各ノードの割当値も計算できる。これを

$V'(0), V'(1), \dots, V'(k-1), V'(k)$  とする。

#### 【0306】

(5-3) 各  $a = a(0), \dots, a(n-1)$  について、 $a$  の  $a_l$  による認証点を  $AP(a, a_l)$  とし、 $k_l = \text{level}(AP(a, a(n)))$  とおく。

#### 【0307】

(5-4) ステップ(3)により、 $a$  の認証パスノードのうちレベルが  $k_l$  より小さいものの割当値は計算済みである。

#### 【0308】

(5-5) また、レベルが  $k_l$  の  $a$  の認証パスノードは  $a_l$  のルートパスに属するノードでレベルが  $k_l$  のものと一致する。従って、このような認証パスノードの割当値は、上記(5-2)で計算した  $V'(k_l)$  となる。

#### 【0309】

(5-6)  $k_l < j < k$  となる  $j$  について、 $a$  の認証パスノードのうちレベルが  $j$  のものは、 $a_l$  の認証パスノードのうちのレベルが  $j$  のものと一致する。従って、その認証パスノードの割当値は、上記(5-1)で計算した、 $V'(j)$  である。

#### 【0310】

以上、(5-1) 及至(5-6)により、各  $a = a(0), \dots, a(n-1)$  について  $a$  の全ての認証パスノードの割当値を計算することが出来る。

#### 【0311】

図39で示す具体例について言えば、インデックス2の登録点(リーフ識別番号5)の認証パスノードは、(0, 4), (1, 3), (2, 0), (3, 1), (4, 1)である。また  $a_l$  をインデックス5の登録点(リーフ識別番号11)とすると、 $AP(a, a_l) = (3, 0)$  であり、 $k_l = \text{level}(AP(a, a_l)) = 3$  である。 $a$  の認証パスノードのうちレベルが  $k_l = 3$  より小さい(0, 4), (1, 3), (2, 0)についてはそれらの割当値は、上記ステップ(5-3)で計算できる(図39、図42参照)。また、レベルが  $k_l = 3$  の認証パスノード(3, 1)の割当値は(5-2)で計算できる(図41、図42参照)。また、レベルが  $k_l = 3$  より大きい(4, 1)についてはステップ(5-1)で計算できる(図41、図42参照)。

以上のステップ(1)及至(5)の手順により、取得した証明書の完全化を行うために、同時にメモリ上に保持する必要があるデータは、間引き抽出データと、一つの局所データのみである。登録点の総数を  $N$  とし、上記ステップ(1)で用いる間引き間隔を  $m$  とすると、同時にメモリ上に保持する必要がある登録点の数は、 $(N/m) + m$  となる。 $m = \sqrt{N}$  とする

と、 $(N/m) + m = 2 \cdot \sqrt{N}$  となり、必要なメモリ容量のオーダーを  $N$  から  $\sqrt{N}$  に削減することが可能となる。

#### 【0312】

<2-5. 利用者装置51の補完データによるルート値計算>

次に、利用者装置51の補完データによるルート値計算について、説明する。これは、利用者装置51の第1の検証機能におけるルート値計算について詳しく説明するものである。

#### 【0313】

ある集約間隔11が終了したとき、利用者装置51は、集約間隔11の間に送信した1つの証明要求R0 に対する受理証明書の個別完全化を上述の方法で実行することにより、完全認証バスデータを計算することができる。完全認証バスデータから、以下に示す(1)乃至(5)に示す手順により、当該の集約間隔の順次集約木のルートの割当値が計算できる。

#### 【0314】

まず完全認証バスデータは、即時補完データと遅延補完データからなる。即時補完データや遅延補完データは、(位置情報、LRタグ、割当値(ハッシュ値))で示す形式の補完データ要素から成るものとする。尚、LRタグはLというタグかRというタグのどちらかの値をとる。ここで、上記の位置情報はレベル情報を含むものとする。レベル情報の間には以下のような2項関係<<が定義されているものとする。

#### 【0315】

任意の登録点について、その完全認証バスデータに含まれる補完データ要素を(位置情報 $P(i)$ 、LRタグ $T(i)$ 、割当値 $H(i)$ ) (但し、 $i = 1, \dots, n$ )とし、位置情報 $P(i)$ に含まれるレベル情報を $level(P(i))$ と表すと、2項関係<<は $level(P(1)), \dots, level(P(n))$ の間に線形順序を定義するものとする。

#### 【0316】

位置情報を1つの集約木におけるレベル(非負整数で表される)とレベル内のインデックスの組み合わせとし、位置情報のうちレベル情報は組み合わせの第1要素とすると、上記2項関係<<としては、整数の大小関係<をとればよい。

#### 【0317】

図43は、完全認証バスデータによる順次集約木のルート値の計算方法を示すフローチャートである。同図によれば、完全認証バスデータのチェック、即ち、即時補完データはLタグ、遅延補完データはRタグを有するかのチェック、及び完全認証バスデータに重なるレベル情報がないかのチェックを確認した後、レベル情報の順序でソートし、各割当値をLRタグに合わせて接続して、ルート値を計算する(ステップS3101, S3102, S3103, S3104, S3105, S3106)。

#### 【0318】

次に、図44及び図45に示すように、ある集約間隔の順次集約木のリーフの1つとして、前の集約間隔の順次集約木のルート値を取り入れる場合について説明する。このような場合には、異なる順次集約木間における受理証明書発行の時間的前後を検証することが簡単にできるという効果がある。例えば、図45において、利用者装置2Aの登録点aが順次集約木ST(5)の部分木ST1(5)のリーフに割り当てられており、また、利用者装置2Bの登録点bが順次集約木ST(6)の部分木ST1(6)のリーフに割り当てられているときは、aとbの合流点がノードR(6)、aのbによる認証点がノードR(5)になるので、登録点aから計算された認証点R(5)の値が、登録点bの即時補完データに含まれていれば、登録点aの登録は登録点bの登録より時間的に前であることを検証することができる。

#### 【0319】

図44は、図45のように複数の順次集約木がリンクされた状況において、1つの順次集約木ST(n)を抽出したものである。

#### 【0320】

ここで、ルート $R(n-1)$ は、順次集約木ST(n-1)と順次集約木ST(n)に共通するノードで

レベル情報 (L, TID(n-1), k(n-1))

レベル内 index 0

位置情報 ((L, TID(n-1), k(n-1)), 0)

である。尚、上述のレベル情報は、

(LRタグ、非負整数の集約木番号、非負整数の集約木内のレベル情報)

で表現されており、LRタグは、LというタグとRというタグのどちらかの値をとる。

#### 【0321】

また、部分木 STl(n) のルート Rl(n)は、

レベル情報 (R, TID(n-1), kl(n))

レベル内 index 1

位置情報 ((R, TID(n-1), kl(n)), 1)

であり、ここで、kl(n) はSTl(n) の高さである。

#### 【0322】

また、部分木STl(n) に関しては、

ルート Rl(n)以外のノードについては、レベル情報 (R, TID(n), j)

ルートRl(n)以外のノードの位置情報は((R, TID(n), j), i) である。

#### 【0323】

ここで、j, i は非負整数であり、leafs(STl(n)) の各要素の位置情報は((R, TID(n), 0), i) と表すことができる。

#### 【0324】

また、R(n)は順次集約木ST(n) とST(n+1)に共通するノードで、

レベル情報 (L, TID(n), k(n))

レベル内 index 0

位置情報 ((L, TID(n), k(n)), 0) (但し、k(n) = kl(n) + 1 とする)

割当値  $V(R(n)) = h(V(R(n-1))) \parallel V(Rl(n))$

である。

また、順次集約小木 STl(n) とルートR(n-1)、 ルートR(n) からなる二分木をST(n) と書く。即ち、

$root(ST(n)) = R(n)$ 、

$leftChild(R(n)) = R(n-1)$ 、

$rightChild(R(n)) = Rl(n)$ 。

#### 【0325】

第n番目の集約期間に対応する順次集約木は ST(n) である。

#### 【0326】

但し、n=0 に対しては、R(n-1) の代わりに所定の割当値を持ったノードIRを用いる (図45参照)。IR の位置情報は ((L, -1, 0), 0) である。

#### 【0327】

このとき、2つの拡張レベル情報の間の順序<<を次のように定義する。

#### 【0328】

$\forall j1, j2, T1, T2 \geq 0 [ (R, T2, j2) << (L, T1, j1) ]$ 、

$\forall j1, j2, T1, T2 \geq 0 [ T1 < T2 \Rightarrow (L, T1, j1) << (L, T2, j2) ]$ 、

$\forall j1, j2, T1 \geq 0 [ j1 < j2 \Rightarrow (R, T1, j1) << (R, T1, j2) ]$ 。

#### 【0329】

この定義により、二項関係<<は任意の登録点の認証パスノードの集合に線形順序を定めるものであることが導かれる。

#### 【0330】

図46は、この定義された2項関係<<を用いて補完データによる集約木ルート値を計算する一例を示すもので、計算は以下のようになる。

#### 【0331】

位置情報 ((R, 10, 0), 5) のノードに対して、即時補完データは、

[ ( (L, 9, k(9)), 0), L, V(R(9))),  
 ( (R, 10, 2), 0), L, V((R, 10, 2), 0)),  
 ( (R, 10, 0), 4), L, V((R, 10, 0), 4) ) ]

となる。ここで (R, 10, 2) << (L, 9, 10) である。

#### 【0332】

遅延補完データは、[ ( (R, 10, 1), 3), R, V((R, 10, 1), 3) ) ]  
 補完データからのルート値の計算を図43のフローに従って行う。

#### 【0333】

- (1) 即時補完データの各要素はLタグを持つことをチェックする→合格
- (2) 遅延補完データの各要素はRタグを持つことをチェックする→合格
- (3) 即時補完データと遅延補完データを合併する

合併結果は、

[ ( (L, 9, k(9)), 0), L, V(R(9))),  
 ( (R, 10, 2), 0), L, V((R, 10, 2), 0)),  
 ( (R, 10, 0), 4), L, V((R, 10, 0), 4) ),  
 ( (R, 10, 1), 3), R, V((R, 10, 1), 3) ) ]

となる。

#### 【0334】

- (4) 合併結果の中に、重なるレベル情報がないことを確認する→合格
- (5) 合併結果を、レベル情報の順序<基準にソートする

ソート結果は、

[ ( (R, 10, 0), 4), L, V((R, 10, 0), 4) ),  
 ( (R, 10, 1), 3), R, V((R, 10, 1), 3) ),  
 ( (R, 10, 2), 0), L, V((R, 10, 2), 0)),  
 ( (L, 9, k(9)), 0), L, V(R(9))) ]

となる。

#### 【0335】

- (6) ステップ(5)のソートの結果を  
 $(J(0), LR(0), V(0)), \dots, (J(k-1), LR(k-1), V(k-1))$

とおき、当該の登録点の登録値を  $V(0)$  とおき、以下のように再帰的に

$W(0), W(1), \dots, W(k-1), W(k)$

を定義する。

#### 【0336】

(i)  $W(0) = V(0)$

(ii)  $LR(j) = L$  のとき、 $W(j+1) = h(V(j) \parallel W(j))$

$LR(j) = R$  のとき、 $W(j+1) = h(W(j) \parallel V(j))$

これに従って計算すると、 $k=4$  であり、 $W(j)$  は以下のように計算できる。

#### 【0337】

$W(0) = V((R, 10, 0), 5).$

$W(1) = h(V((R, 10, 0), 4) \parallel V((R, 10, 0), 5)).$

$W(2) = h(W(1) \parallel V((R, 10, 1), 3)).$

$W(3) = h(V((R, 10, 2), 0) \parallel W(2)).$

$W(4) = h(V(R(9)) \parallel W(3))$

となる。 $W(3) = V(R(10))$ 、 $W(4) = V(R(10))$  である。

#### 【0338】

従って、第2の実施の形態のイベント順序証明システム200によれば、第1の実施の形態と同じ効果を得ることができる。即ち、木構造を用いてイベント順序を証明するイベント順序証明システムにおいて、利用者装置51から証明要求を受付けた証明装置4が、該証明要求に対して受理証明書を含む連鎖補完方式（登録点に関して、登録点の即時補完データ及び直前登録点の登録点における遅延補完データを証明応答データに含む）による証明

応答を発行したときでも、利用者装置21がこの証明応答を用いて、インクリメンタル完全化を行うと、利用者装置21間における受理証明書発行の時間的前後を検証することができるので、証明要求をまとめた公表データが電子的に公表される前であっても、受理証明書の正当性を検証することができる。

#### 【0339】

また、連鎖補完方式には、シーケンス補完方式よりも、証明応答のデータ量が少なくて済むという効果がある。さらには、連鎖補完方式においても、証明装置4は、順次集約木そのものを記憶部に記憶させる方法は勿論、スタック構造を用いた記憶方法も用いることができるので、証明装置4の必要記憶容量を大幅に減少させることもできる。

#### 【0340】

また、利用者装置51におけるインクリメンタル完全化処理においても、個別完全化及び一括完全化の双方を具備するので、状況に応じて最適なインクリメンタル完全化を行うことにより、受理証明書の正当性を検証することができる。さらに、証明応答データすべてを利用者装置51のメモリ上に記憶させず、部分的な局所データだけを記憶させる方式であっても、インクリメンタル完全化を行うことができるので、利用者装置51の必要なメモリ量を大幅に減少させることができる。

#### 【0341】

また、順次集約期間が終了後においては、利用者装置51は、インクリメンタル完全化の処理により、完全補完データを取得できるので、順次集約木のルート値を計算することができる。さらに、前の集約間隔の順次集約木のルート値を次の順次集約木のリーフの割当値とする場合には、順次集約木をまたがった受理証明書発行の時間的前後を検証することが簡単にできる。

#### 【0342】

以上、本発明の実施例について説明してきたが、本発明の要旨を逸脱しない範囲において、本発明の実施例に対して種々の変形や変更を施すことができる。例えば、上記実施の形態においては、順次集約木として二分木を用いたが、本発明は二分木に限定されるものではなく、1つの親が複数の子を持つ有向木であればよいものである。

#### 【0343】

また、利用者装置21又は51は、一定時間間隔終了前に、証明装置1又は4が運用を中断、あるいは順次集約木のルート値を計算するのに必要なデータを消失したとき、証明装置1又は4の運用中断あるいはデータ消失の時点までに受信し記憶した順序証明応答から、計算可能な割当値を持つ順次集約木のノードのうちで、その親のノードの割当値が計算できないような1つあるいは複数のノードの位置情報と割当値を、電子的に公表する利用者サイド電子的情報公表手段を有してもよい。そして、この場合、所定の検証機関が、この公表情報が矛盾しないことを検証するようにしてもよい。

#### 【0344】

##### < 順次集約木の性質 >

上記実施の形態で用いた順次集約木の性質について詳しく説明する。

#### 【0345】

順次集約木のリーフ番号*i*について、*i*で識別される順次集約木のリーフに順次割当値を割り当てる元となった証明要求を受付けて該リーフに割当値を割り当てる一連の処理を該リーフに対する処理ラウンドと言いround(*i*)と表す。

#### 【0346】

今、甲を利用者装置、乙を監査装置とし、*i*<sub>0</sub>と*i*<sub>1</sub>を*i*<sub>0</sub> < *i*<sub>1</sub>なる二つの順次集約木リーフ番号とし、round(*i*<sub>0</sub>)において甲は受理証明書を受信し、乙はround(*i*<sub>1</sub>)において監査用受理証明書を受信したものとする。このとき、*i*<sub>0</sub>の*i*<sub>1</sub>による認証点は以下の性質を持つ。

#### 【0347】

(1) 認証点の割当値は、監査点、即ちノード(0, *i*<sub>1</sub>)の即時補完データに含まれる。

#### 【0348】

(2) 上記認証点を  $(j', i')$  とおき、 $\text{round}(il)$  終了時にリーフ  $(0, i_0)$  が属する順次集約小木を  $ST_2$  とし、 $(0, i_0)$  の  $ST_2$  における認証パスを  $\text{authPath}_{ST_2}(0, i_0)$  としたとき、 $\text{authPath}_{ST_2}(0, i_0)$  に属するノードで、レベルが  $j'$  より小さいものに対する割当値は、ノード  $(0, i_0)$  に対応するラウンドで受理証明書を受理した利用者が、ノード  $(0, i_1)$  に対応するラウンド以降において受信できる遅延補完データあるいは受信した即時補完データに含まれる。

【0349】

即ち、 $i_1 \leq i_2$  とすると、 $\text{authPath}_{ST_2}(0, i_0)$  に属するノードで、レベルが  $j'$  より小さいものに対する割当値は、 $\text{immedData}(i_0)$  あるいは  $\text{lateData}(i_0, i_2)$  に含まれる。

【0350】

(3)  $ST_2$  におけるリーフ  $(0, i_0)$  のルート・パスを  $\text{rtPath}_{ST_2}(0, i_0)$  としたとき、上記認証点の割当値及びに  $\text{rtPath}_{ST_2}(0, i_0)$  属するノードでレベルが該認証点のレベルより小さいノードの割当値は、ノード  $(0, i_0)$  で受理証明書を受理した利用者が、ノード  $(0, i_1)$  に対応するラウンド以降において受信する遅延補完データおよびノード  $(0, i_0)$  で受信した受理証明書（即時補完データを含む）から計算することができる。

【0351】

(性質の証明)

以下では、利用者に渡す受理証明書に、即時補完データを含める場合について説明する。利用者に渡す受理証明書に即時補完データを含めず、その代わりに遅延補完データにこの情報を含める場合でも同様の議論により同じ結論が得られる。

【0352】

(1) まず、項目(1)について図47、図48を用いつつ説明する。

【0353】

(場合1) 最初に、図47を参照して、 $i_0$  と  $i_1$  が  $i_1$  時点における順次集約フォレスト内の1つの順次集約小木  $ST_2$  に属する場合を考える。ここで、合流点を  $(j, i)$ 、そのレフト・チャイルドである認証点を  $(j', i')$  とおく。ノード  $(0, i_1)$  の順次集約小木  $ST_2$  におけるルート・パス  $\text{rtPath}_{ST_2}(0, i_1)$  において、 $(0, i_1)$  から出発して、合流点に至る直前のノードを  $(j'', i'')$  とおく。このとき、認証点は、 $(j'', i'')$  の左補完点である。従って、認証パス  $\text{authPath}_{ST_2}(i_1)$  の定義から、 $((j', i'), L)$  はノード  $(0, i_1)$  の  $ST_2$  における認証パスに含まれる。また、ノード  $(j', i')$  への値の割当ては、 $\text{round}(i_1)$  より前に終了している。よって、 $((j', i'), L, V(j', i'))$  は  $(0, i_1)$  に対する即時補完データに含まれる。

【0354】

(場合2) 次に、図48を参照して、 $i_0$  と  $i_1$  が  $i_1$  時点における順次集約フォレスト内のどの順次集約小木にも同時には属さない場合を考える。このとき、 $i_0$  は  $i_1$  時点における順次集約フォレスト内のある順次集約小木  $ST_2'$  に属する。このとき、登録点  $(0, i_1)$  に対する即時補完データの定義により、 $V(\text{root}(ST_2'))$  は、登録点  $(0, i_1)$  に対する即時補完データに含まれる。

【0355】

(2) 次に項目(2)について図49乃至52を用いつつ説明する。

【0356】

(場合1) 最初に、図49及び50を参照して、 $i_0$  と  $i_1$  が  $i_1$  時点における順次集約フォレスト内の1つの順次集約小木  $ST_2$  に属する場合を考える。

【0357】

$k = \text{height}(ST_2)$  とおく。

【0358】

認証点  $(j', i')$  はノード  $(0, i_0)$  のルート・パス  $\text{rtPath}_{ST_2}(0, i_0)$  に含まれる。ここで、

$$\text{rtPath}_{ST_2}(0, i_0) = [(0, r(0)), \dots, (j', r(j')), (j'+1, r(j'+1)), \dots, (k, r(k))]$$

とする。また、 $\text{authPath}_{ST_2}(0, i_0)$  の要素で、レベルが  $j'$  より小さいノードの並びを  
[  $(0, s(0)), \dots, (j'-1, s(j'-1))$  ]

とおく。各  $j_1 \in [0..j'-1]$  に対して、 $V(j_1, r(j_1))$  が  $\text{immedData}(i_0)$  あるいは  $\text{lateData}(i_0, i_2)$  に含まれることを示せばよい。

#### 【0359】

$\text{authPath}_{ST_2}(0, i_0)$  の定義により、 $\text{authPath}_{ST_2}(0, i_0)$  のレベル  $j_1$  の要素  $p_2 = (j_1, s(j_1))$  は、 $\text{rtPath}_{ST_2}(0, i_0)$  のレベル  $j_1+1$  の要素  $p_3 = (j_1+1, r(j_1+1))$  のライト・チャイルドであるか或いはレフト・チャイルドである。どちらであるかによって場合分けする。

#### 【0360】

(場合1-1)  $p_2$  が  $p_3$  のライト・チャイルドであるとき、図49に示すように、 $p_2$  の割当値  $V(p_2)$  は、 $i_1 \leq i_2$  なる  $i_2$  において、甲が受信できる遅延補完データ  $\text{lateData}(i_0, i_2)$  に含まれる。なぜならば、リーフ  $(0, i_1)$  に対応するラウンドのイベント順序証明処理が終わった時点で、図49のBで表された SB2 の部分木の割当値は計算可能であり計算され割当て済みである。従って、その時点以降で発行される登録点  $i_0$  に対する遅延補完データにはBのルート  $p_2$  の割当値  $V(p_2)$  が含まれるからである。

#### 【0361】

(場合1-2)  $p_2$  が  $p_3$  のレフト・チャイルドであるとき、図50に示すように、ノード  $p_2$  の割当値  $V(p_2)$  は、登録点  $i_0$  に対する即時補完データに含まれる。何故ならば、図50の  $p_1$  をルートとする部分木Bについて、

$$\forall i \in \text{leaves}(B) [ i < i_0 ]$$

であり、従って  $i_0$  で識別されるラウンドの開始時に、 $\text{leaves}(B)$  の割当値は確定している。よって、 $p_2 = \text{root}(B)$  の割当値は、 $i_0$  で識別されるラウンドにおいて確定しており、従って  $p_2$  は  $i_0$  時点において値が確定している  $i_0$  の認証パスノードの集合に含まれるからである。

#### 【0362】

(場合2) 次に、図51及び図52を参照して、 $i_0$  と  $i_1$  が  $i_1$  時点における順次集約フォレスト内のどの順次集約小木にも同時には属さない場合を考える。このとき、 $i_0$  は  $i_1$  時点における順次集約フォレスト内のある順次集約小木  $ST_3$  に属し、 $\text{root}(ST_3)$  が  $i_0$  の  $i_1$  による認証点となる。 $k = \text{height}(ST_3)$  と置く。認証点  $(j', i')$  は  $(0, i_0)$  のルートパス  $\text{rtPath}_{ST_3}(0, i_0)$  に含まれる。ここで、

$$\text{rtPath}_{ST_3}(0, i_0) = [(0, r(0)), \dots, (j', r(j')), (j'+1, r(j'+1)), \dots, (k, r(k))]$$

とする。また、 $\text{authPath}_{ST_3}(0, i_0)$  の要素で、レベルが  $j'$  より小さいノードの並びを

$$[(0, s(0)), \dots, (j'-1, s(j'-1))]$$

とおく。各  $j_1 \in [0..j'-1]$  に対して、 $V(j_1, r(j_1))$  が  $\text{immedData}(i_0)$  あるいは  $\text{lateData}(i_0, i_2)$  に含まれることを示せばよい。

#### 【0363】

$\text{authPath}_{ST_3}(0, i_0)$  の定義により、 $\text{authPath}_{ST_3}(0, i_0)$  のレベル  $j_1$  の要素  $p_2 = (j_1, s(j_1))$  は、 $\text{rtPath}_{ST_3}(0, i_0)$  のレベル  $j_1+1$  の要素  $p_3 = (j_1+1, r(j_1+1))$  のライト・チャイルドであるか或いはレフト・チャイルドである。どちらであるかによって場合分けする。

#### 【0364】

(場合2-1)  $p_2$  が  $p_3$  のライト・チャイルドであるとき、図51に示すように、 $p_2$  の割当値  $V(p_2)$  は、 $i_1 \leq i_2$  なる  $i_2$  において、甲が受信できる遅延補完データ  $\text{lateData}(i_0, i_2)$  に含まれる。なぜならば、リーフ  $(0, i_1)$  に対応するラウンドのイベント順序証明処理が終わった時点で、図51のBで表された SB3 の部分木の割当値は計算可能であり計算され割当て済みであり、従ってその時点以降で発行される登録点  $i_0$  に対する遅延補完データにはBのルート  $p_2$  の割当値  $V(p_2)$  が含まれるからである。

#### 【0365】



(場合 2-2)  $p_2$ が $p_3$ のレフト・チャイルドであるとき、図52に示すように、ノード $p_2$ の割当値 $V(p_2)$ は、登録点 $i_0$ に対する即時補完データに含まれる。何故ならば、図52の $p_1$ をルートとする部分木 $B$ について、

$$\forall i \in \text{leaves}(B) [i < i_0]$$

であり、従って $i_0$ で識別されるラウンドの開始時に、 $\text{leaves}(B)$ の割当値は確定している。よって、 $p_2 = \text{root}(B)$ の割当値は、 $i_0$ で識別されるラウンドにおいて確定しており、従って $p_2$ は $i_0$ 時点において値が確定している $i_0$ の認証パスノードの集合に含まれるからである。

【0366】

(3) 認証パスの定義及び項目(2)から、各 $j_1 \in [0..j']$ に対して、 $V(j_1, r(j_1))$ を以下のように再帰的に計算することが出来る。

【0367】

まず、 $V(j_1, r(j_1))$ は受理証明書に含まれているノード $(0, i_0)$ の割当値とする。

【0368】

次に、 $j_1 \in [0..j'-1]$ に対して、 $V(j_1, r(j_1))$ が計算されたと仮定し、 $V(j_1+1, r(j_1+1))$ を以下のように計算する。 $r(j_1) < s(j_1)$ のときは、

$$V(j_1+1, r(j_1+1)) = h(V(j_1, r(j_1)) \parallel V(j_1, s(j_1)))$$

とし、 $s(j_1) < r(j_1)$ のときは、

$$V(j_1+1, r(j_1+1)) = h(V(j_1, s(j_1)) \parallel V(j_1, r(j_1)))$$

とする。

【図面の簡単な説明】

【0369】

【図1】本発明の第1の実施の形態に係るイベント順序証明システムのシステム構成図である。

【図2】本発明の第1の実施の形態に係るイベント順序証明システムに用いられる順次集約木の構成を説明する図である。

【図3】本発明の第1の実施の形態に係るイベント順序証明システムのイベント順序受理証明書の構成を示す図である。

【図4】本発明の第1の実施の形態に係るイベント順序証明システムにおいて各登録点とその補完データを説明する図である。

【図5】本発明の第1の実施の形態に係るイベント順序証明システムの利用者装置におけるイベント順序の判定方法を説明する図である。

【図6】本発明の第1の実施の形態に係るイベント順序証明システムのイベント順序証明方法の動作を説明するシーケンス図である。

【図7】本発明の第1の実施の形態に係るイベント順序証明システムのイベント順序証明検証方法の動作を説明するシーケンス図である。

【図8】本発明の第1の実施の形態に係るイベント順序証明システムのイベント順序証明検証方法の動作を説明するシーケンス図である。

【図9】本発明の第2の実施の形態に係るイベント順序証明システムのシステム構成図である。

【図10】本発明の第2の実施の形態に係るイベント順序証明システムのイベント順序受理証明書の構成を示す図である。

【図11】本発明の第2の実施の形態に係るイベント順序証明システムにおける完全化波及処理を説明する図である。

【図12】本発明の第2の実施の形態に係るイベント順序証明システムにおいて完全化波及処理を用いることにより、連鎖補完方式の証明応答からシーケンス補完方式の証明応答を計算できることを示す図である。

【図13】本発明の第2の実施の形態に係るイベント順序証明システムにおいて第1の連鎖補完方式を説明する図である。

【図14】本発明の第2の実施の形態に係るイベント順序証明システムにおいて第1

の連鎖補完方式による証明応答作成の動作を説明するフローチャートである。

【図15】本発明の第2の実施の形態に係るイベント順序証明システムにおいて第2の連鎖補完方式を説明する図である。

【図16】本発明の第2の実施の形態に係るイベント順序証明システムにおいて第2の連鎖補完方式による証明応答作成の動作を説明するフローチャートである。

【図17】本発明の第2の実施の形態に係るイベント順序証明システムの第2の連鎖補完方式におけるデータ構造の一例である。

【図18】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式における即時補完データ及び遅延補完データの計算手順の一例を説明するフローチャートである。

【図19】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるノード値計算順の一例を説明するフローチャートである。

【図20】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式における遅延データ設定手順の一例を説明するフローチャートである。

【図21】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式における順次集約木の切替処理の一例を説明するフローチャートである。

【図22】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式における順次集約木の終端・切替処理のサブルーチンの一例を説明するフローチャートである。

【図23】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式における順次集約木の切替処理のサブルーチンの一例を説明するフローチャートである。

【図24】図22の処理を具体的に説明する図である。

【図25】順次集約フォレストと順次集約小木を説明する図である。

【図26】順次集約フォレストと現時点順次集約木を説明する図である。

【図27】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル完全個別化の動作を説明するフローチャートである。

【図28】図27の処理を具体的に説明する図である。

【図29】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル完全個別化の順次集約小木決定の計算手順の一例を説明するフローチャートである。

【図30】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル個別完全化の取得参照点決定の計算手順の一例を説明するフローチャートである。

【図31】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル個別完全化の連鎖補完データ蓄積用データ構造の一例である。

【図32】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル個別完全化のアルゴリズムを説明する図である。

【図33】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル個別完全化の認証パスノードの割当値の計算手順の一例を説明するフローチャートである。

【図34】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル個別完全化の認証パスノードの各割当値の計算手順の一例を説明するフローチャートである。

【図35】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル一括個別化の手順の一例を説明するフローチャートである。

【図36】本発明の第2の実施の形態に係るイベント順序証明システムの連鎖補完方式におけるインクリメンタル一括個別化の根拠を説明する図である。

【図 3 7】本発明の第 2 の実施の形態に係る イベント 順序 証明 システム の 連鎖 補完 方式における インクリメンタル一括個別化の根拠を説明する図である。

【図 3 8】本発明の第 2 の実施の形態に係る イベント 順序 証明 システム の 連鎖 補完 方式における インクリメンタル 完全化（一部をメモリに納め、多段式に実行する方式）を説明する図である。

【図 3 9】本発明の第 2 の実施の形態に係る イベント 順序 証明 システム の 連鎖 補完 方式における インクリメンタル 完全化（一部をメモリに納め、多段式に実行する方式）を説明する図である。

【図 4 0】本発明の第 2 の実施の形態に係る イベント 順序 証明 システム の 連鎖 補完 方式における インクリメンタル 完全化（一部をメモリに納め、多段式に実行する方式）を説明する図である。

【図 4 1】本発明の第 2 の実施の形態に係る イベント 順序 証明 システム の 連鎖 補完 方式における インクリメンタル 完全化（一部をメモリに納め、多段式に実行する方式）を説明する図である。

【図 4 2】本発明の第 2 の実施の形態に係る イベント 順序 証明 システム の 連鎖 補完 方式における インクリメンタル 完全化（一部をメモリに納め、多段式に実行する方式）を説明する図である。

【図 4 3】完全 認証 バスデータによる 順次 集約 木の ルート 値 の 計算 方法 を 示す フローチャートである。

【図 4 4】ある 集約 間隔 の 順次 集約 木の リーフ の 1 つ として、前の 集約 間隔 の 順次 集約 木の ルート 値 を 取り 入れる 場合 の 順次 集約 木 を 説明 する 図 である。

【図 4 5】ある 集約 間隔 の 順次 集約 木の リーフ の 1 つ として、前の 集約 間隔 の 順次 集約 木の ルート 値 を 取り 入れる 場合 の 順次 集約 木 を 説明 する 図 である。

【図 4 6】ある 集約 間隔 の 順次 集約 木の リーフ の 1 つ として、前の 集約 間隔 の 順次 集約 木の ルート 値 を 取り 入れる 場合 の 完全 認証 バスデータによる 順次 集約 木の ルート 値 の 計算 方法 を 説明 する 図 である。

【図 4 7】認証 点 の 割当 値 は、監査 点 の 受理 証明 書 内 補完 データ に 含まれる こと を 説明 する 図 である。

【図 4 8】認証 点 の 割当 値 は、監査 点 の 受理 証明 書 内 補完 データ に 含まれる こと を 説明 する 図 である。

【図 4 9】認証 点 の レベル より 低い 認証 バス ノード は、遅延 補完 データ あるいは 受理 証明 書 内 補完 データ に 含まれる こと を 説明 する 図 である。

【図 5 0】認証 点 の レベル より 低い 認証 バス ノード は、遅延 補完 データ あるいは 受理 証明 書 内 補完 データ に 含まれる こと を 説明 する 図 である。

【図 5 1】認証 点 の レベル より 低い 認証 バス ノード は、遅延 補完 データ あるいは 受理 証明 書 内 補完 データ に 含まれる こと を 説明 する 図 である。

【図 5 2】認証 点 の レベル より 低い 認証 バス ノード は、遅延 補完 データ あるいは 受理 証明 書 内 補完 データ に 含まれる こと を 説明 する 図 である。

【図 5 3】イベント 順序 証明 システム の 概念 を 説明 する 図 である。

【図 5 4】線形 リンキング を 用いた イベント 順序 証明 システム の 概念 を 説明 する 図 である。

#### 【符号の説明】

【0370】

- 1, 4 … イベント 順序 証明 装置
- 21, 51 … イベント 順序 証明 利用者 装置
- 3 … コンピュータ ネットワーク
- 11, 21 … 送受信部
- 12 … イベント 順序 証明 要求 集約 部
- 13, 41 … イベント 順序 証明 応答 作成 部
- 14 … デジタル 署名 作成 部

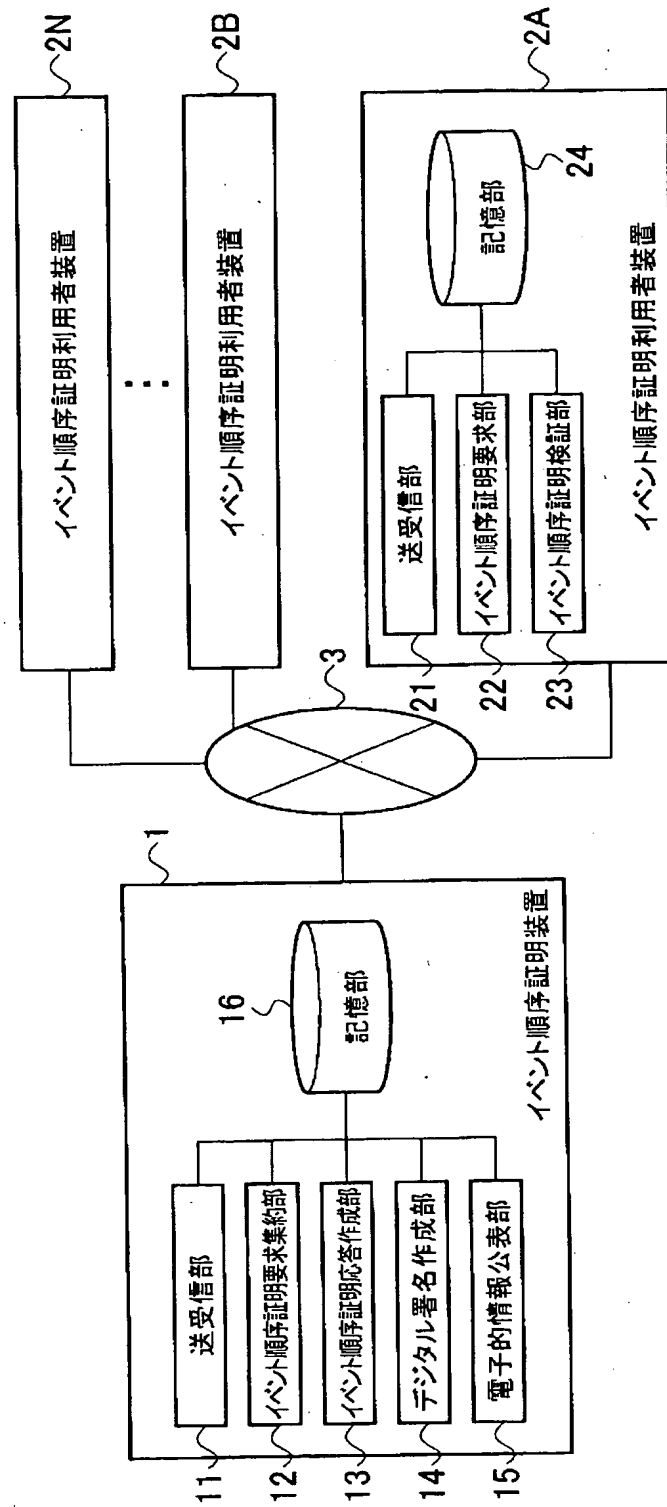
1 5 … 電子的情報公表部

1 6 , 2 4 , 4 2 … 記憶部

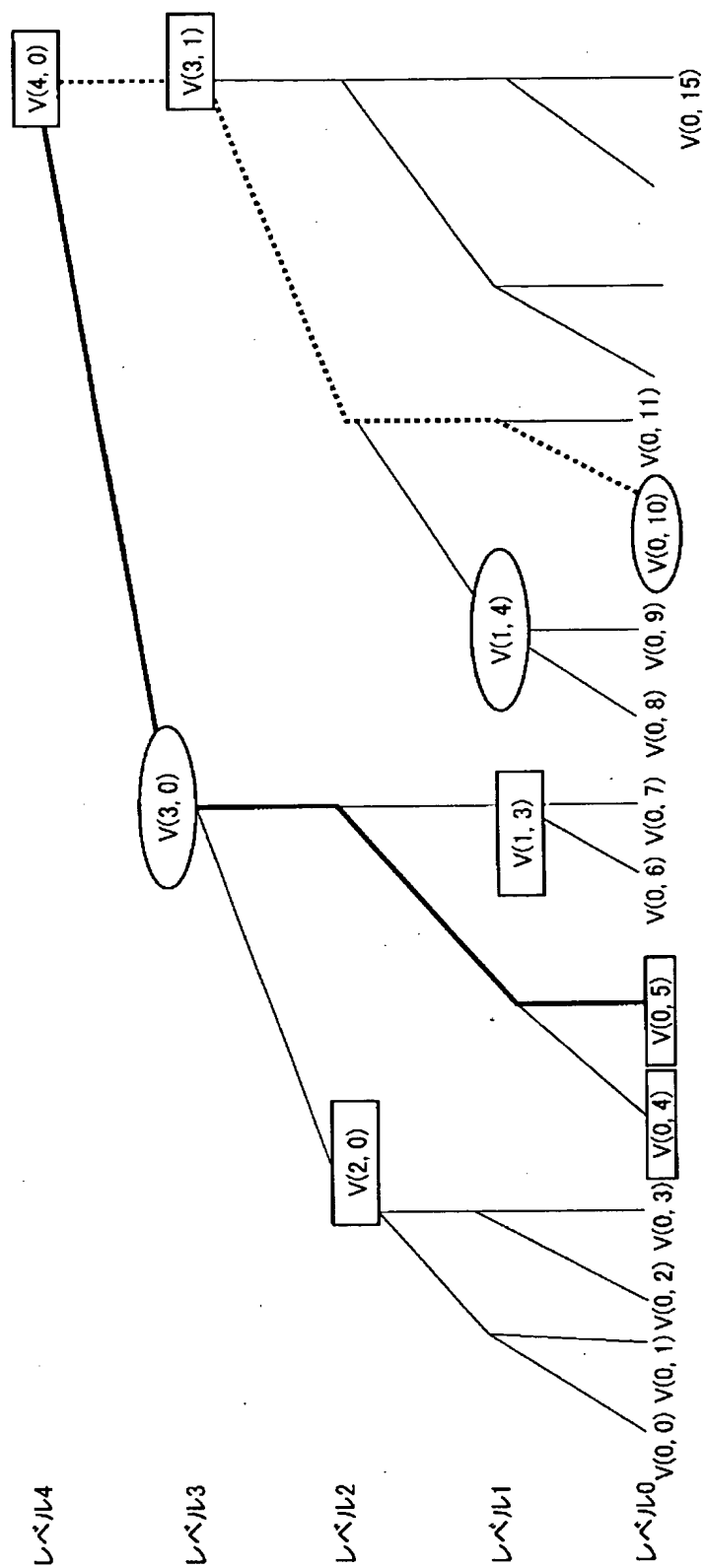
2 2 … イベント順序証明要求部

2 3 , 5 1 … イベント順序証明検証部

1 0 0 , 2 0 0 … イベント順序証明システム



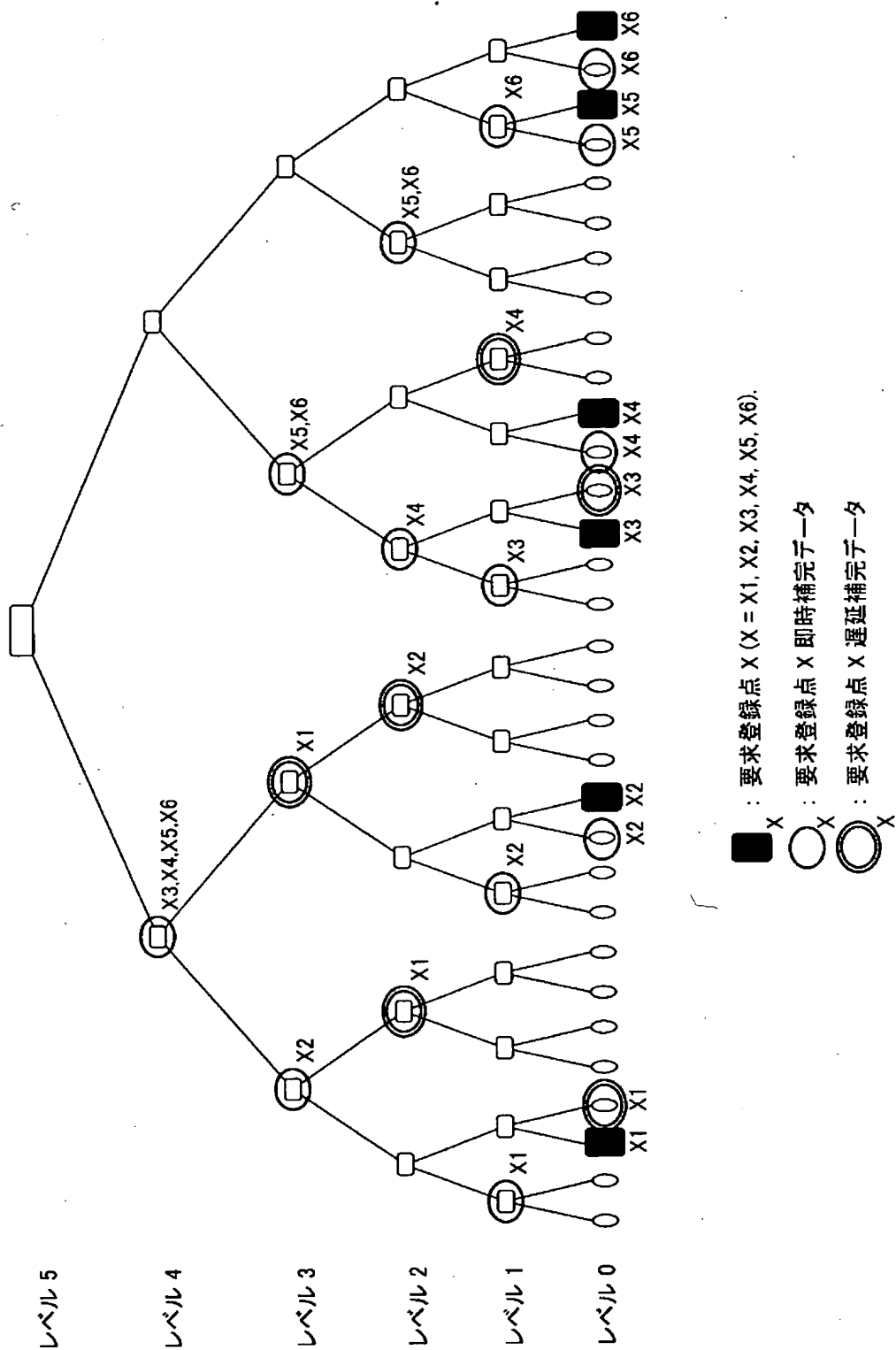
100 イベント順序証明システム



【図 3】

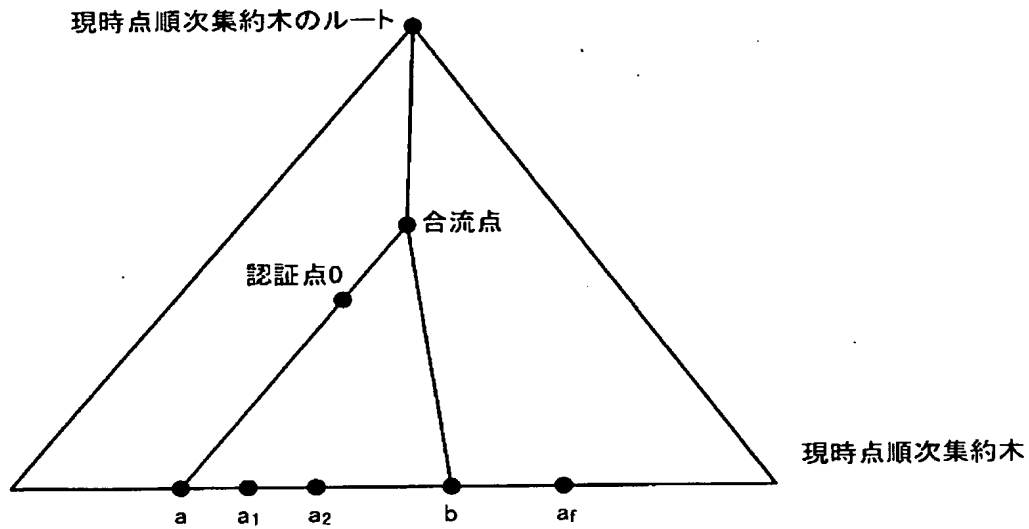
項目	記号	必須
元デジタルデータ	y	○
順次割当データ	z	○
順次集約木番号	n	○
順次集約木リーフ番号	i	○
登録点の即時補完データ(位置情報、割当値)	SK	○
過去の各登録点の遅延補完データ(位置情報、割当値)	TK	○

イベント順序受理証明書  
EOC(y)

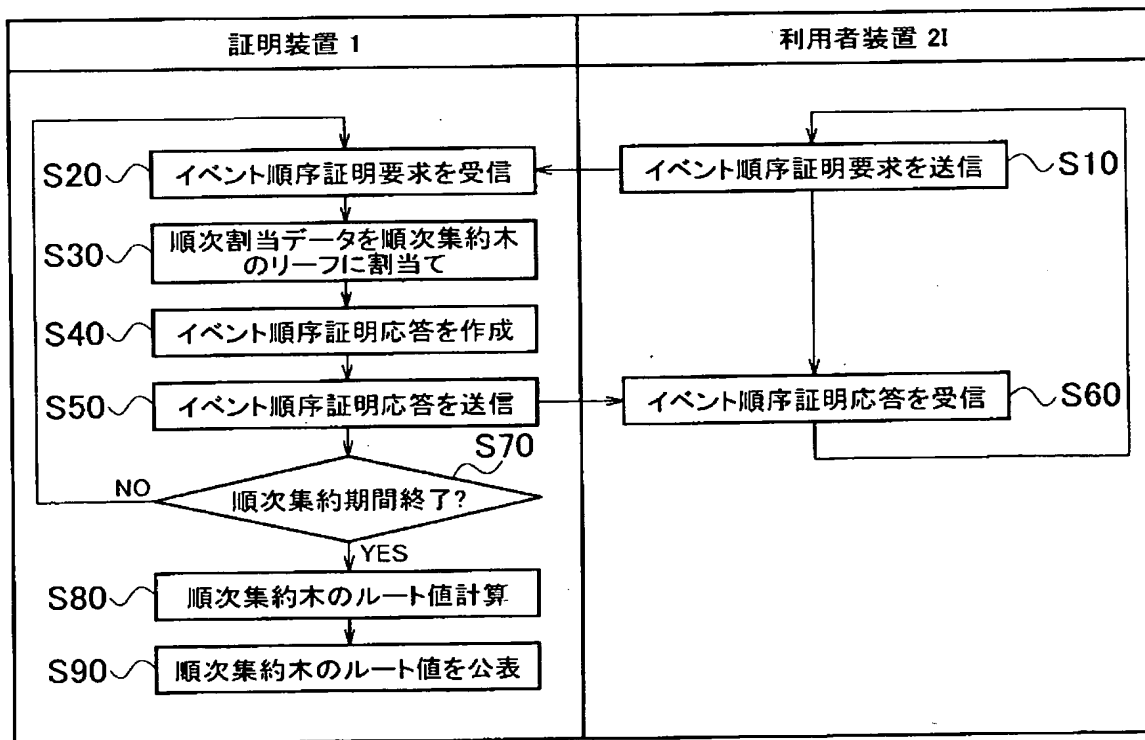


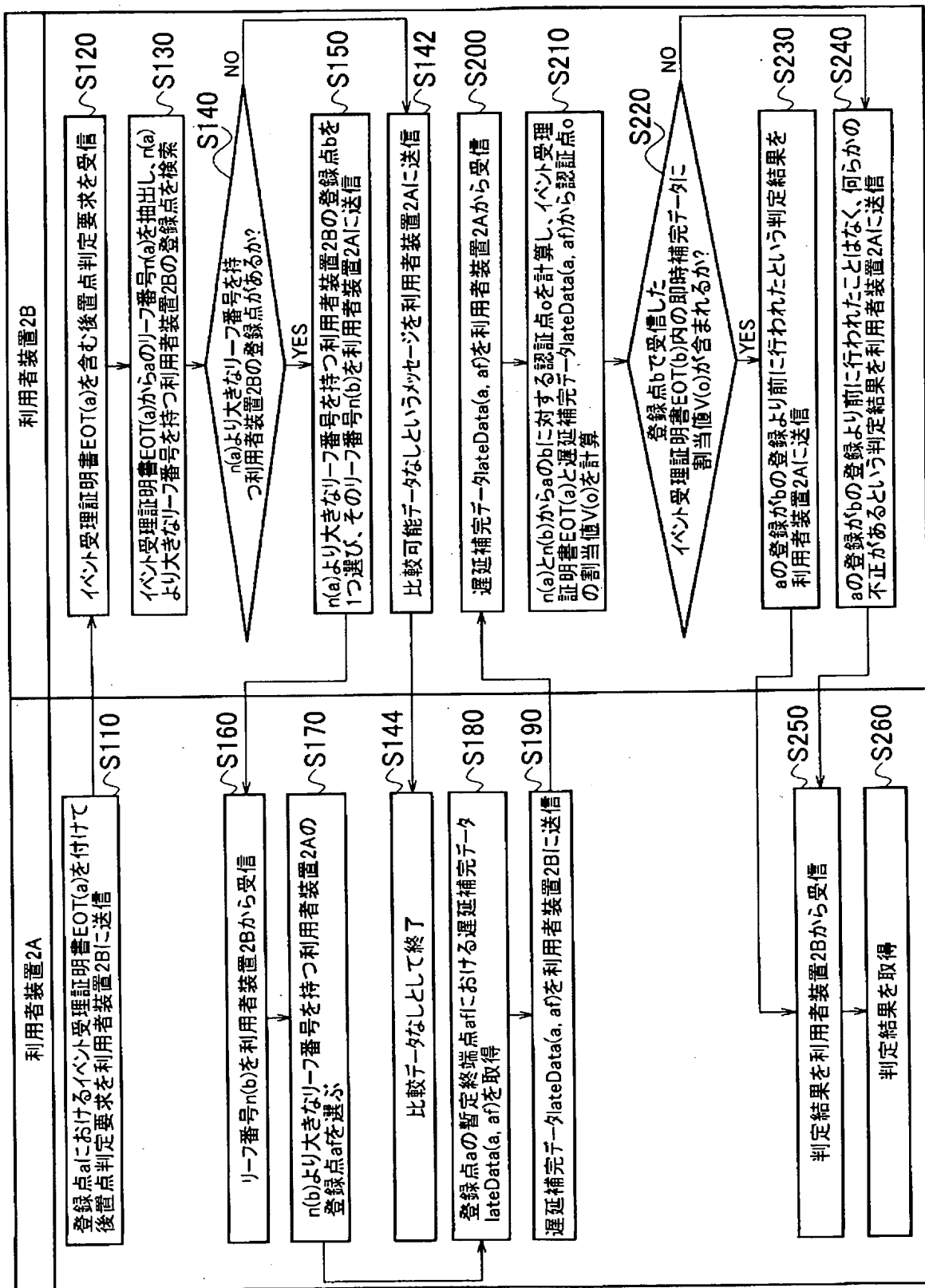


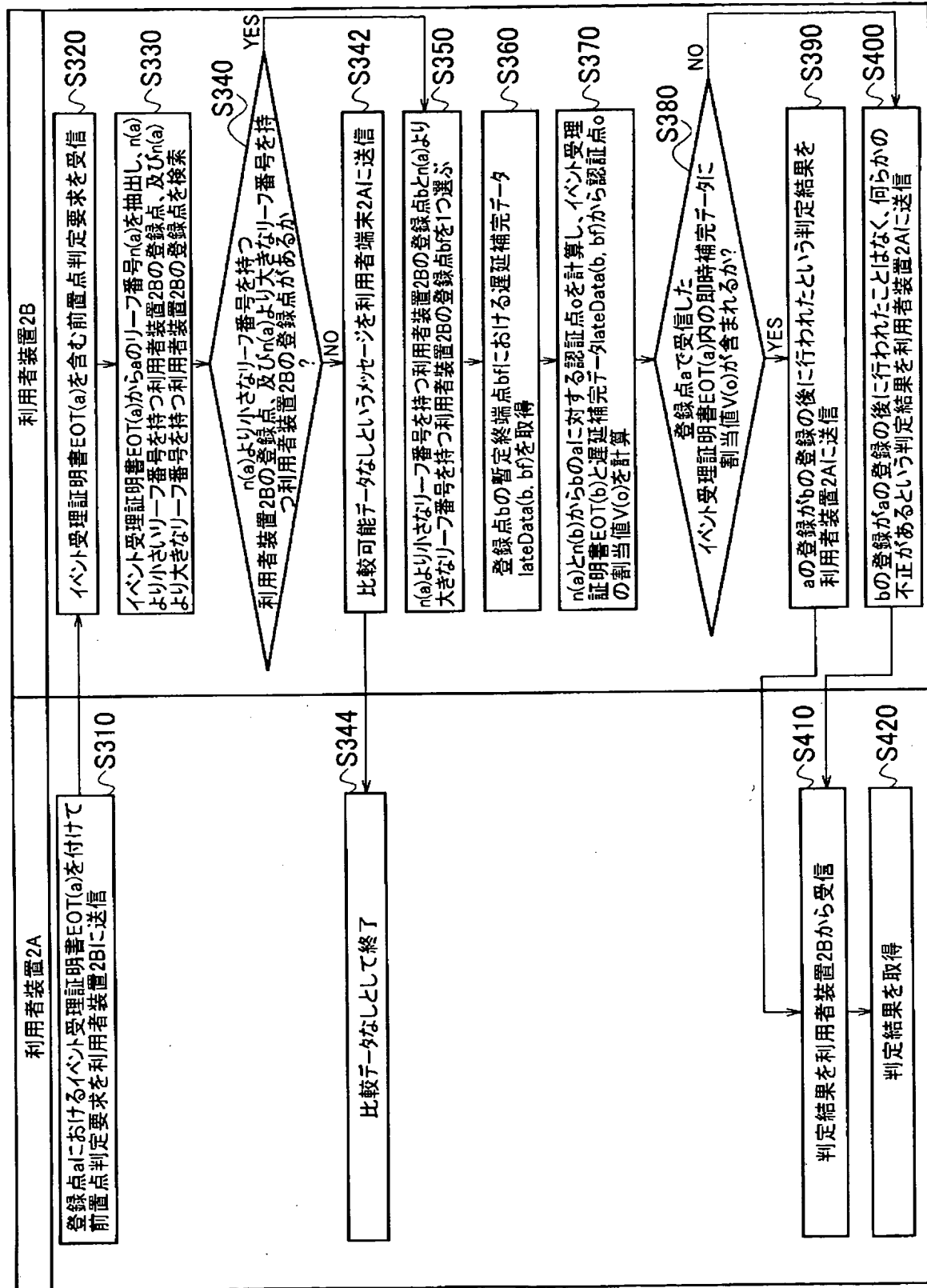
【図 5】

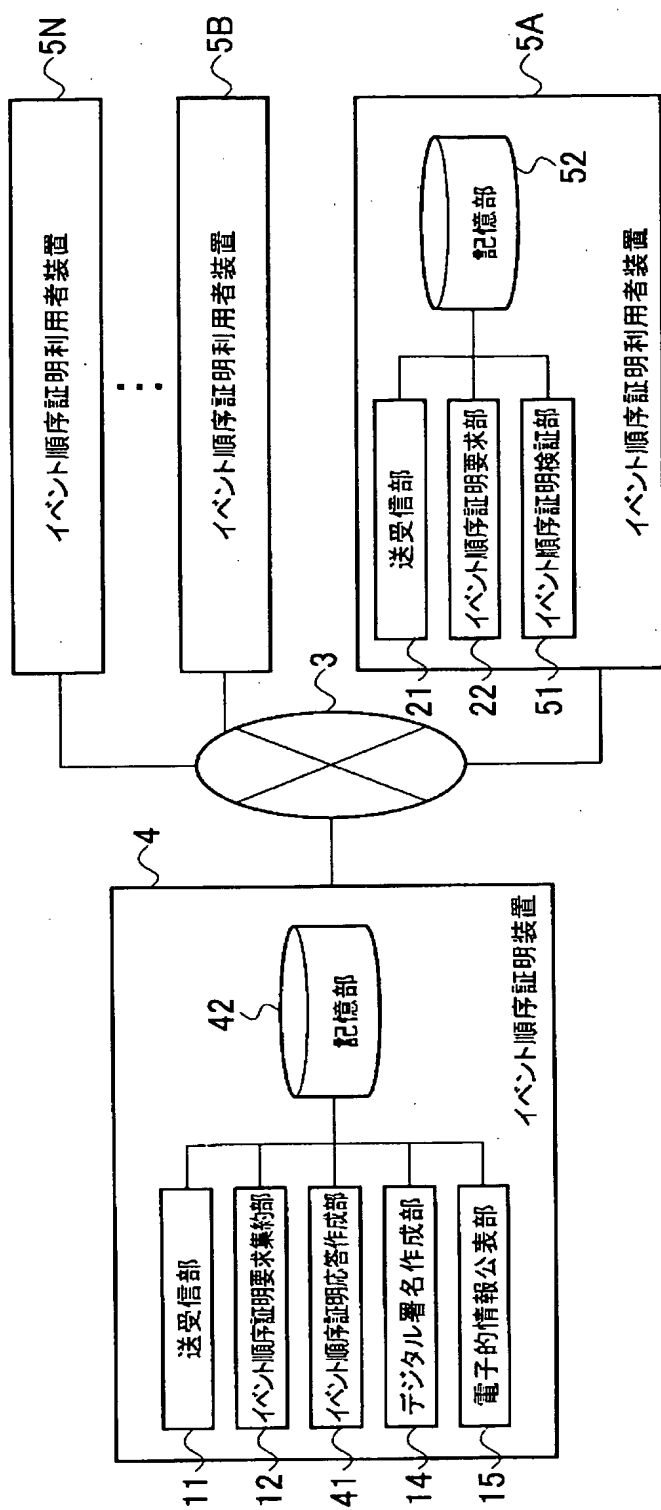


【図 6】







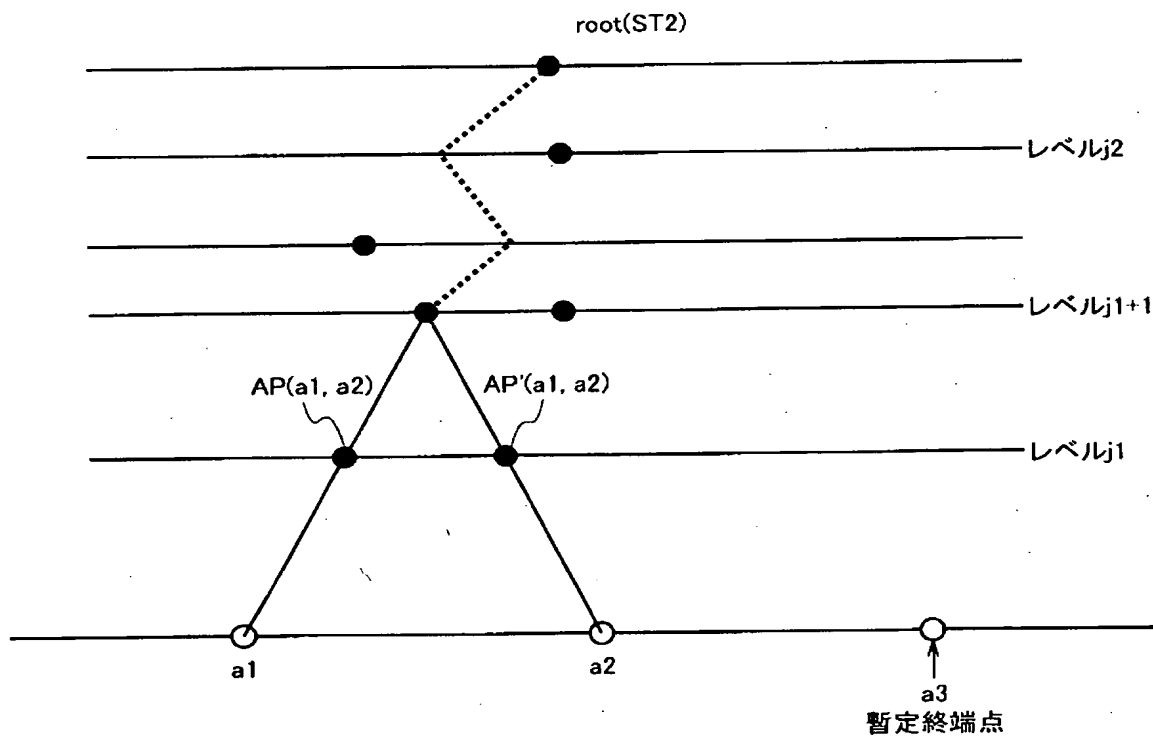


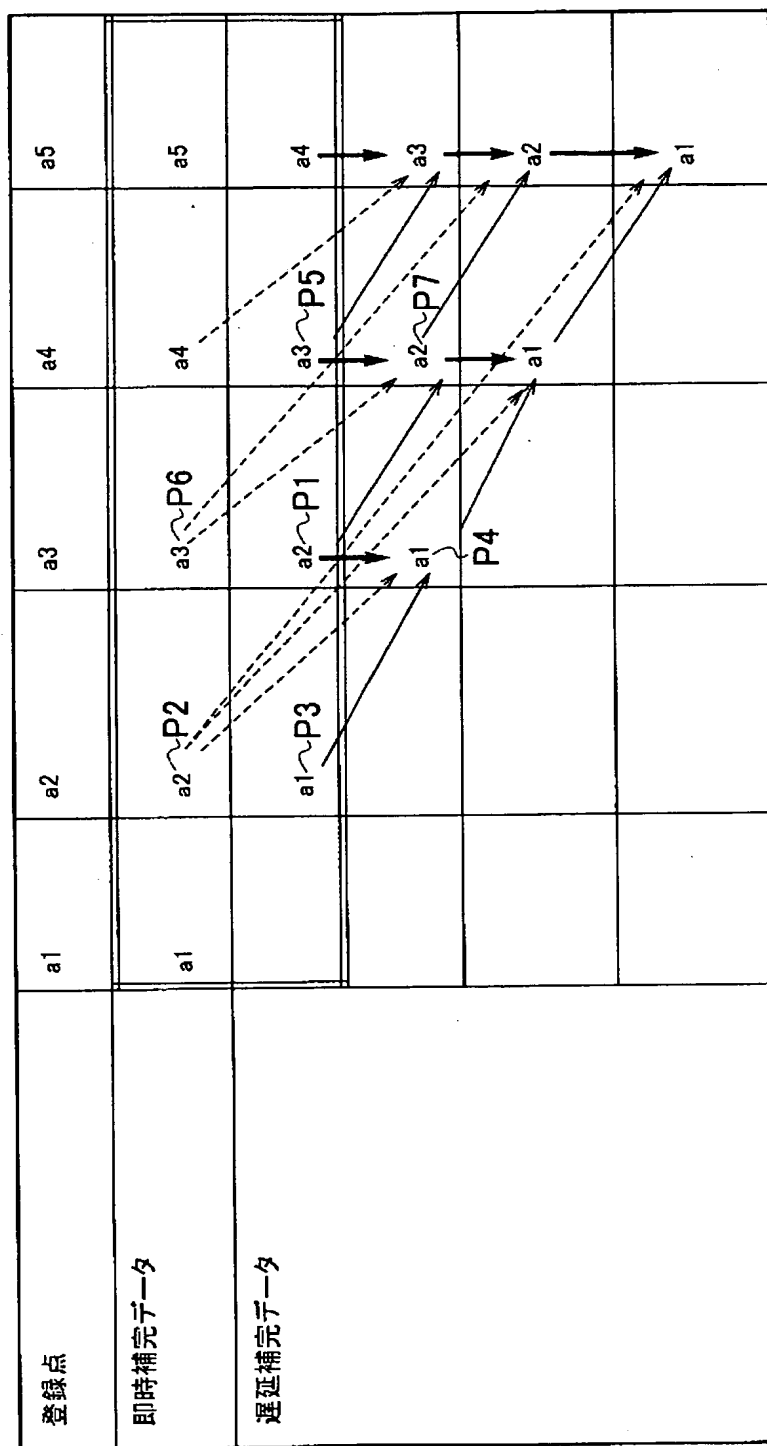
200 イベント順序証明システム

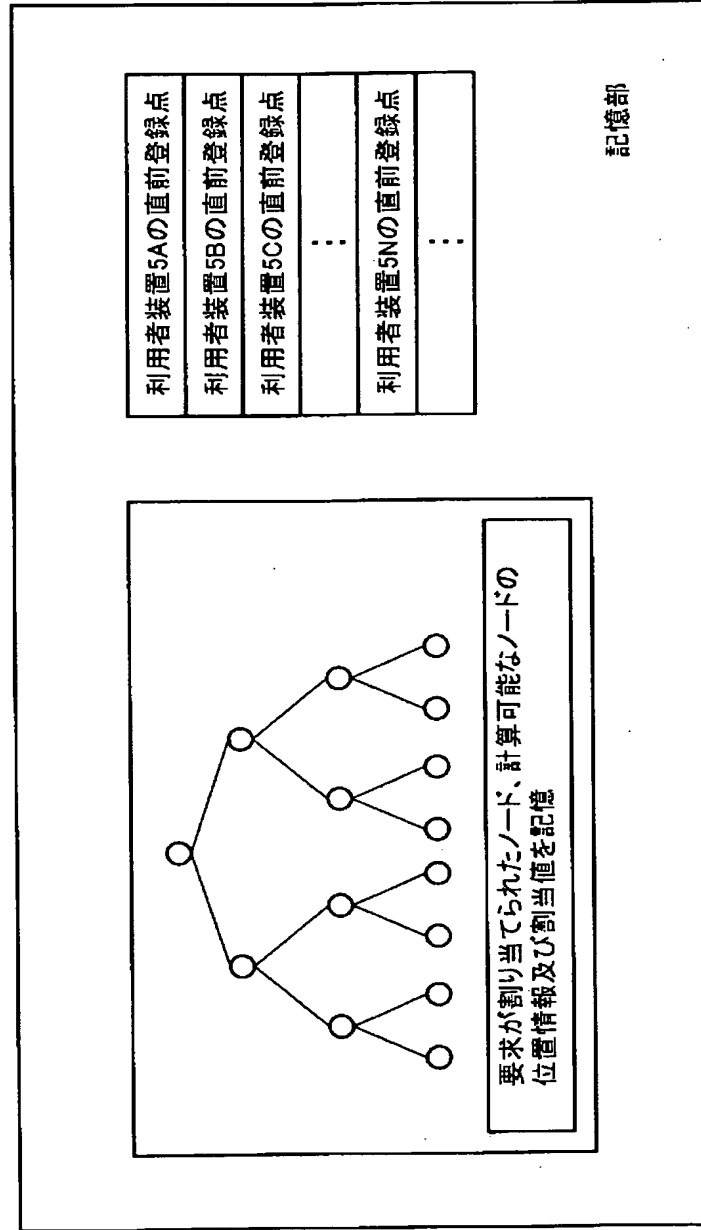
【図 1 0】

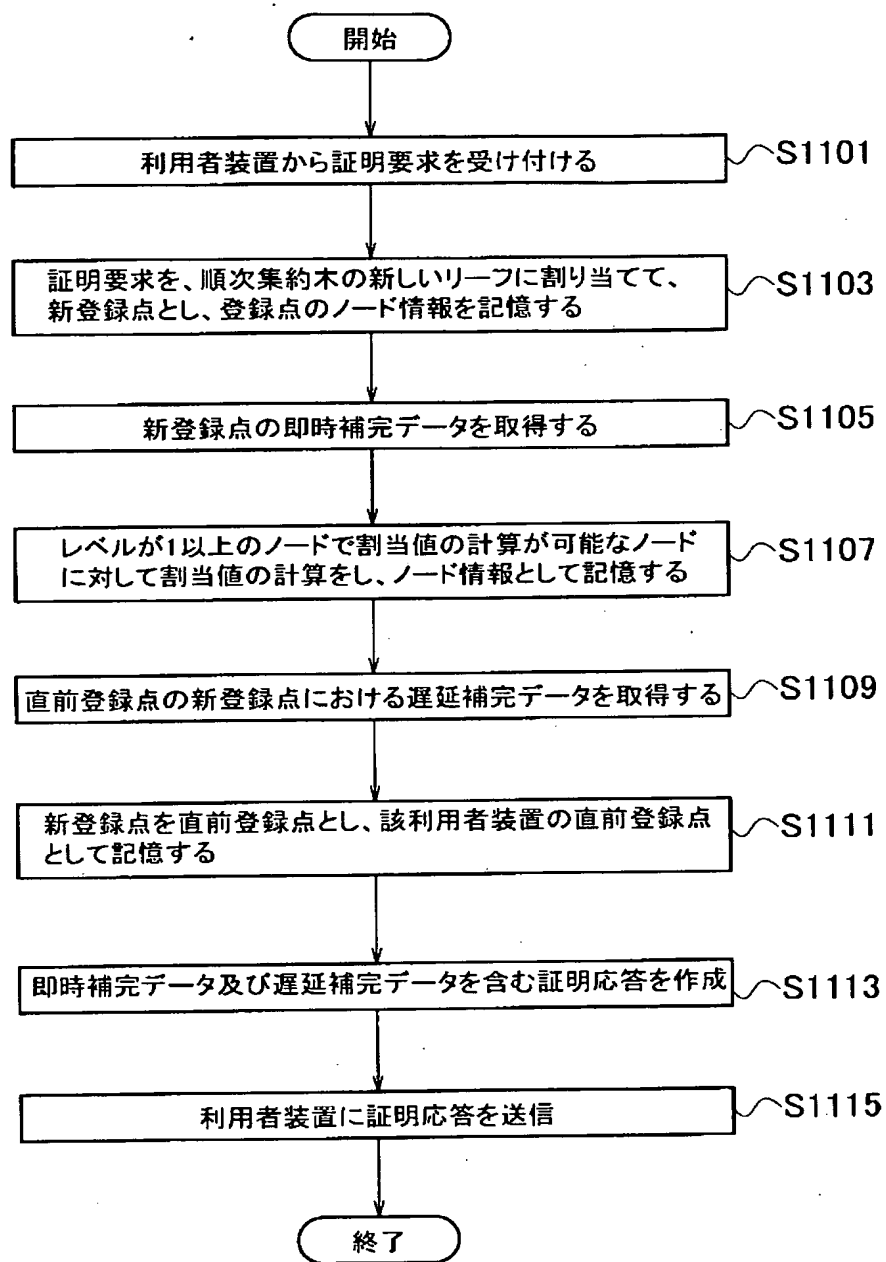
項目	記号	必須	イベント順序受理証明書 EOC(y)
元デジタルデータ	y	○	
順次割当データ	z	○	
順次集約木識別番号	n	○	
順次集約木リーフ番号	i	○	
登録点の即時補完データ(位置情報、割当値)	SK	○	
直前登録点の遅延補完データ(位置情報、割当値)	TK2	○	

【図 1 1】

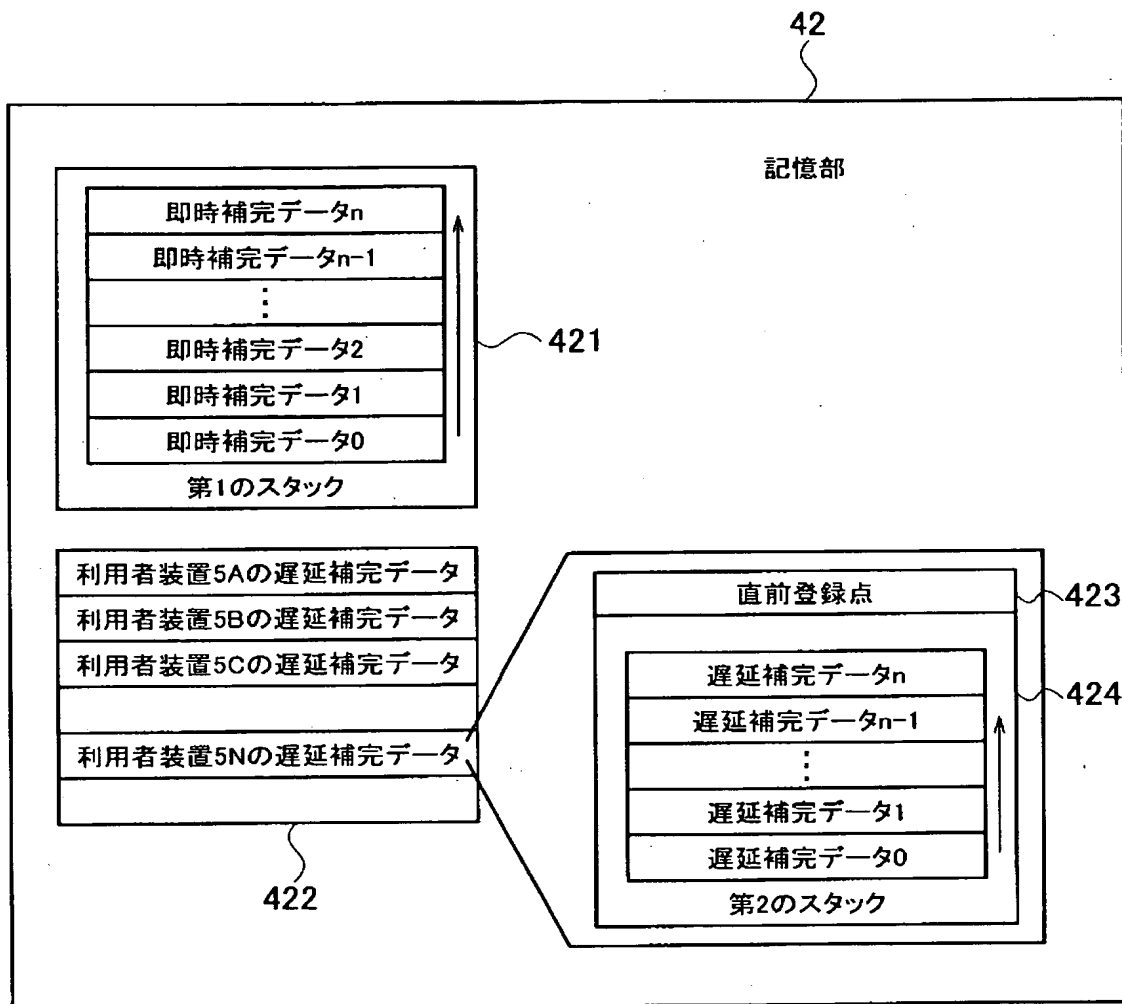


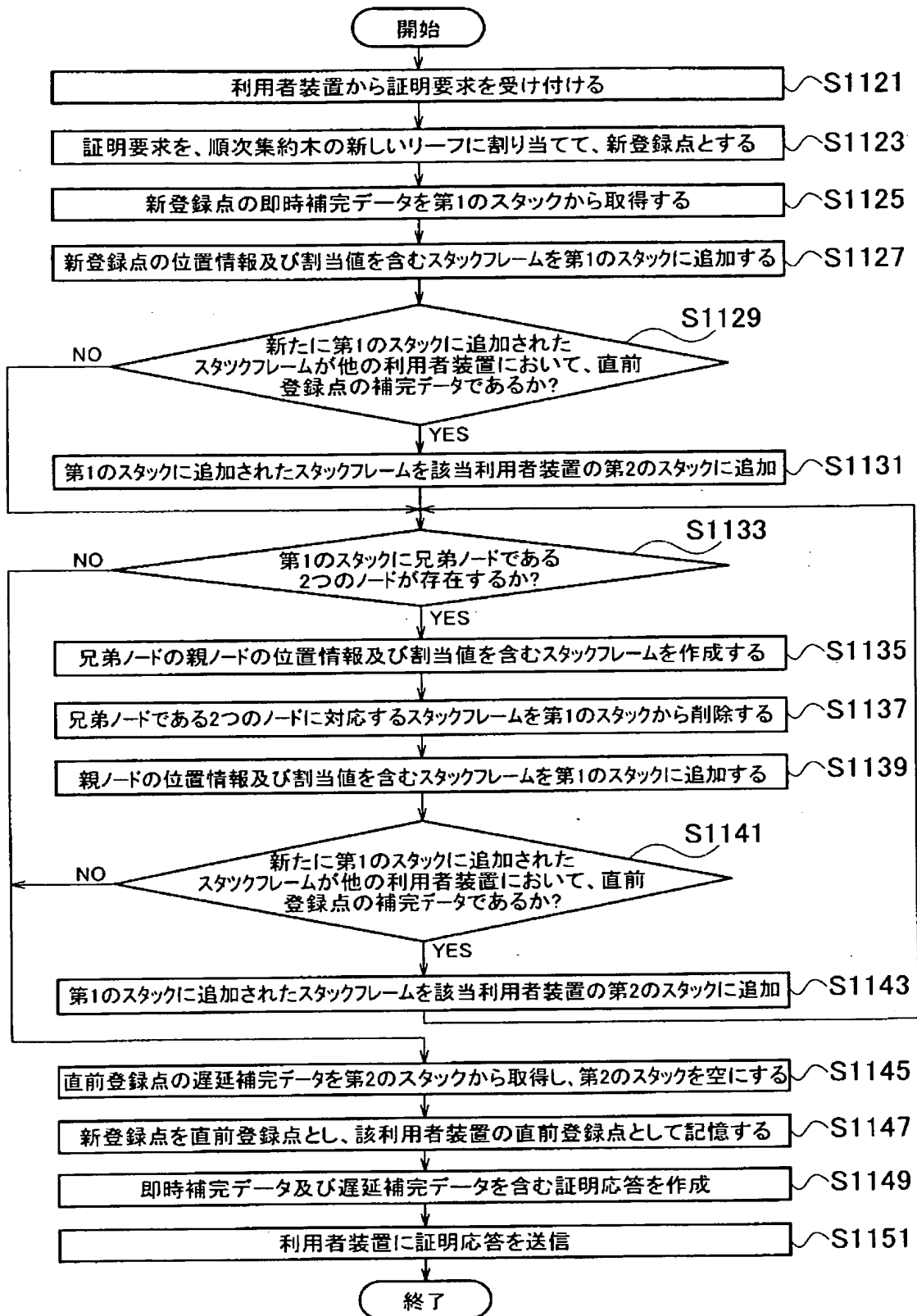


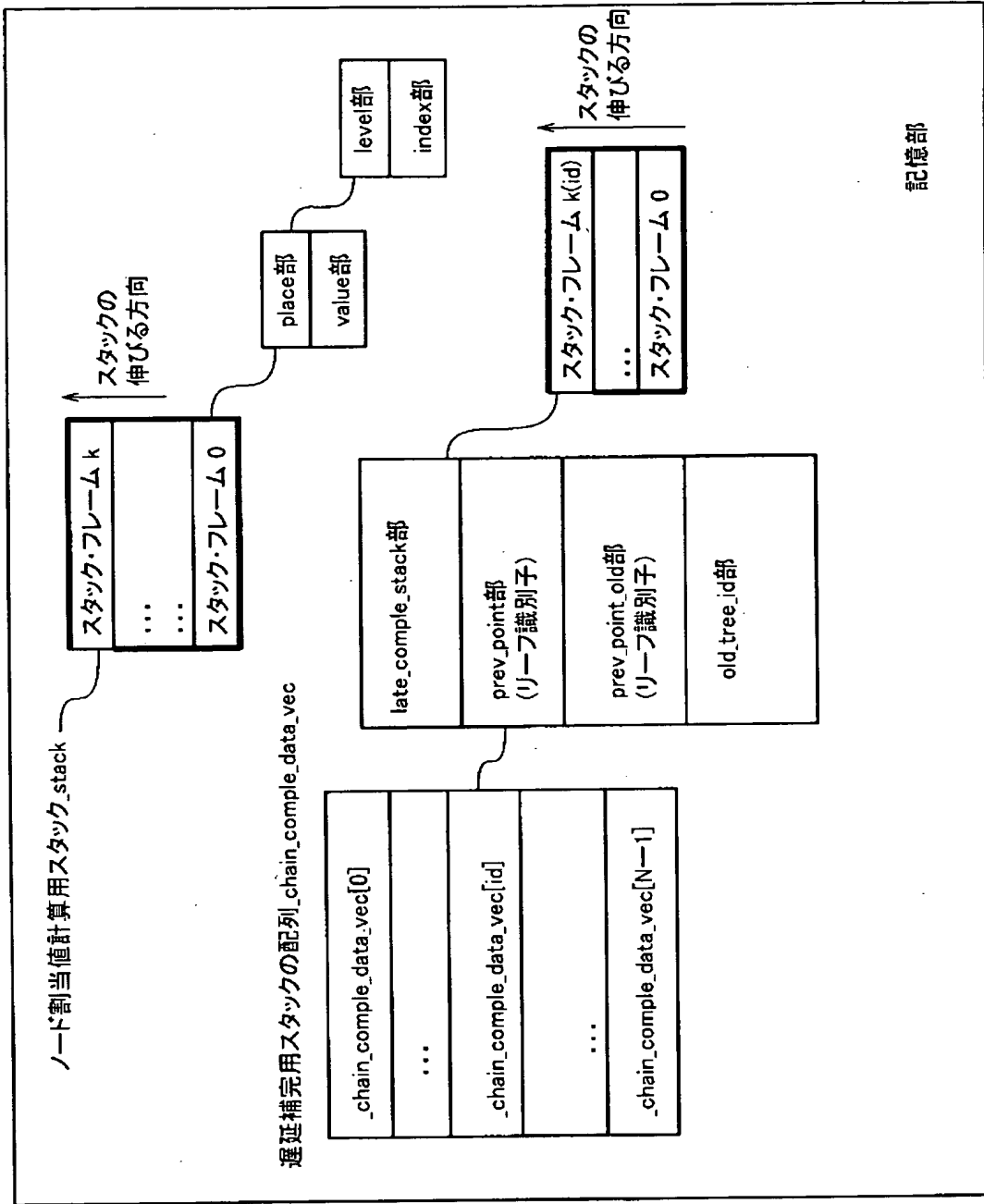


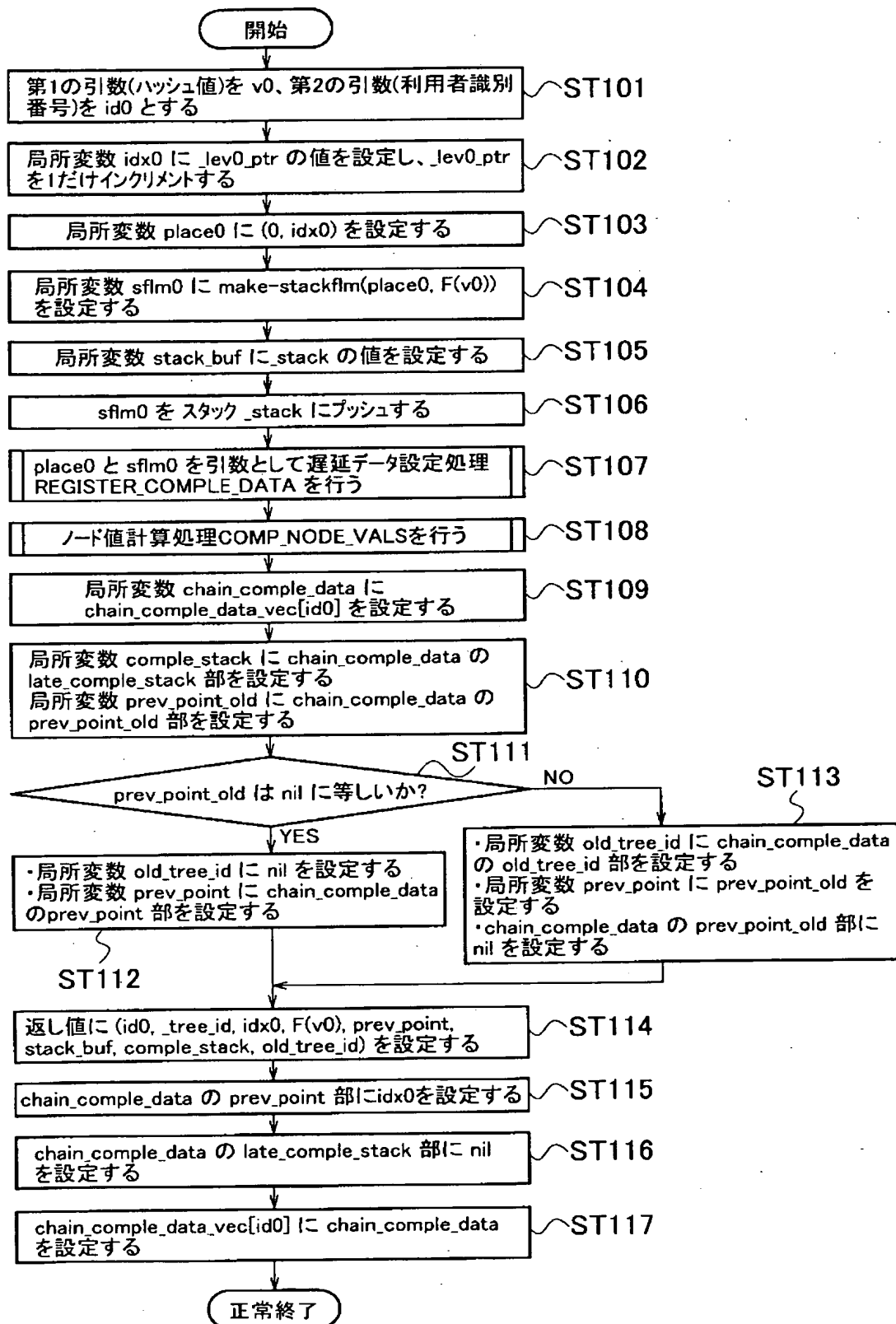


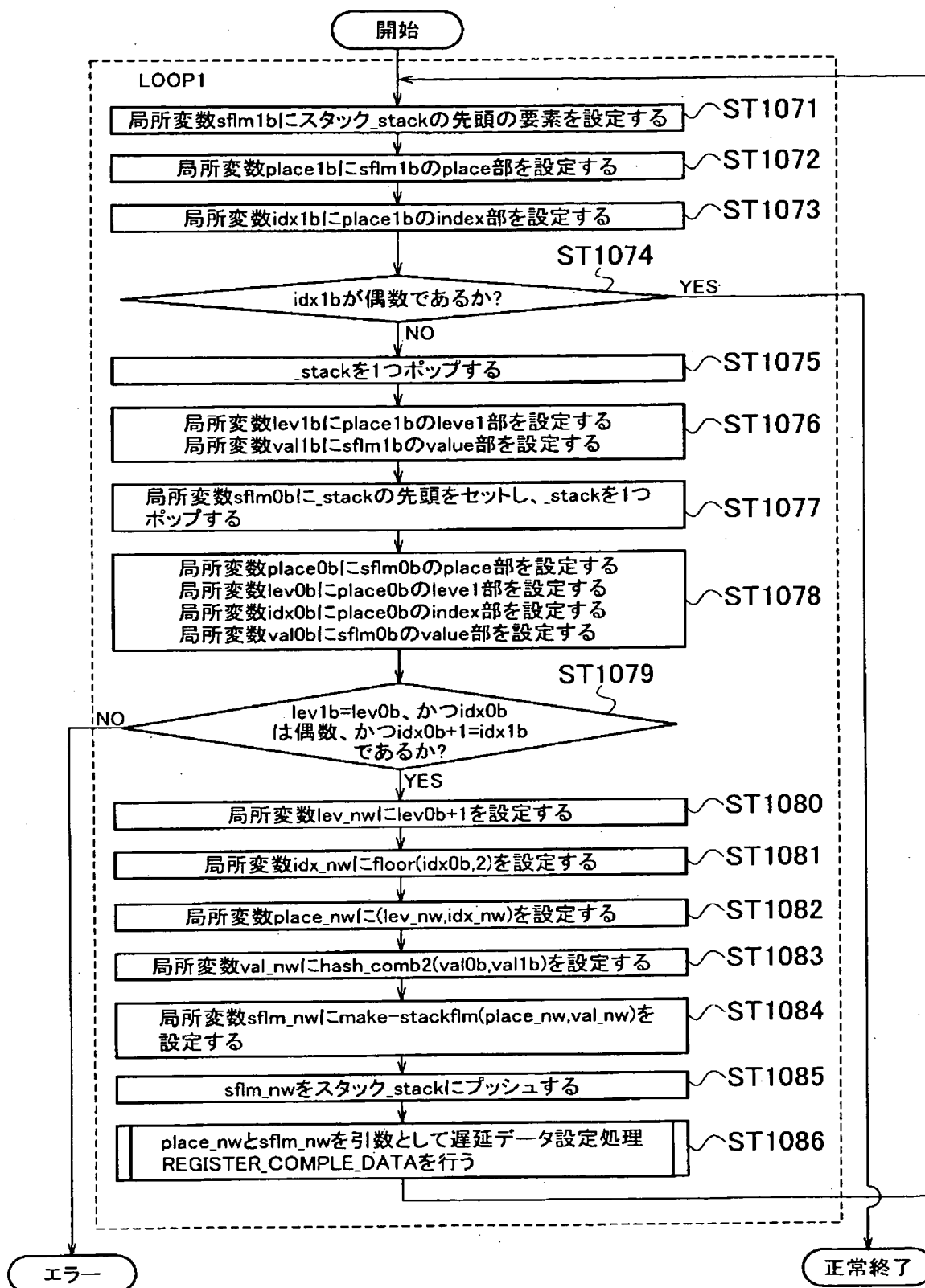




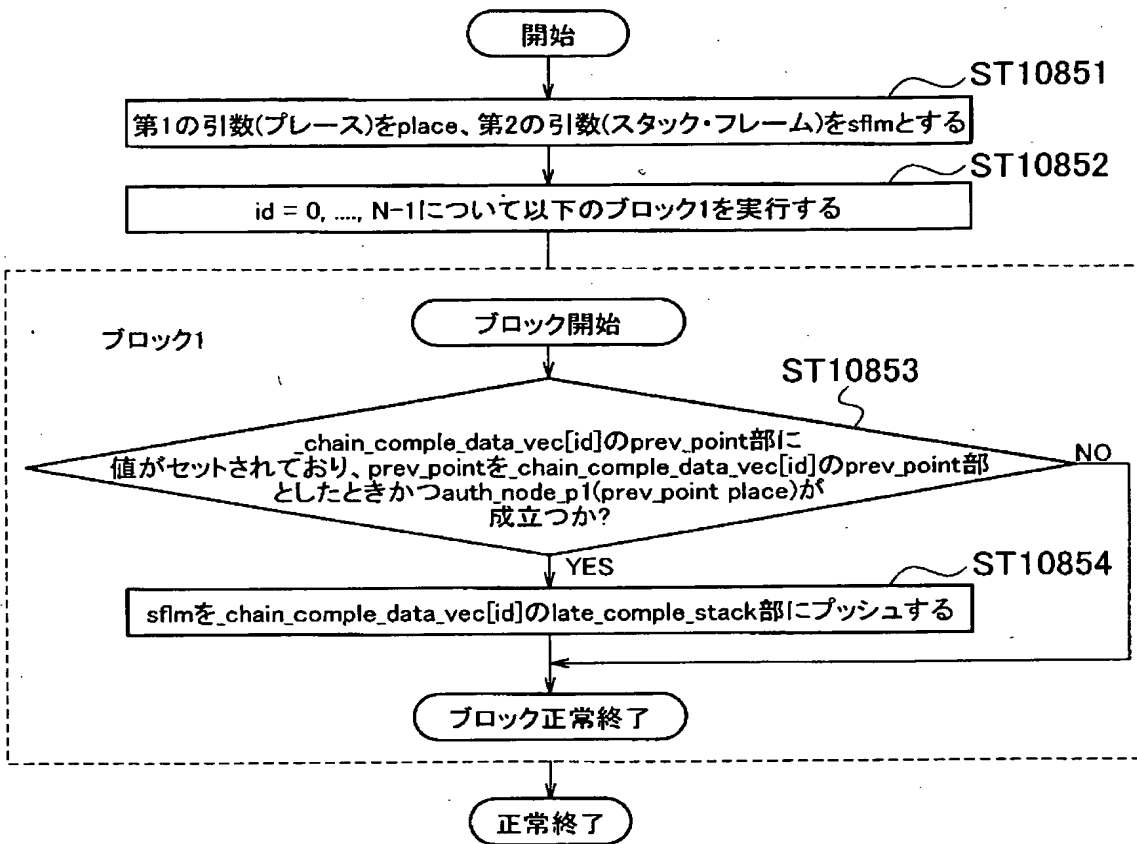




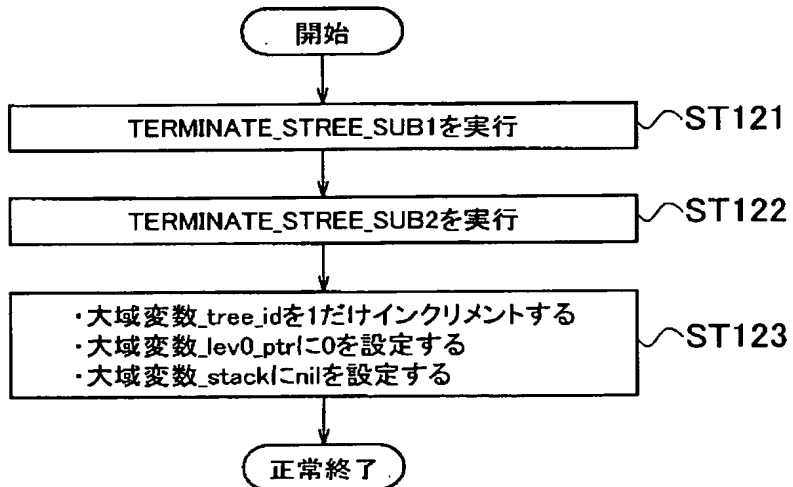




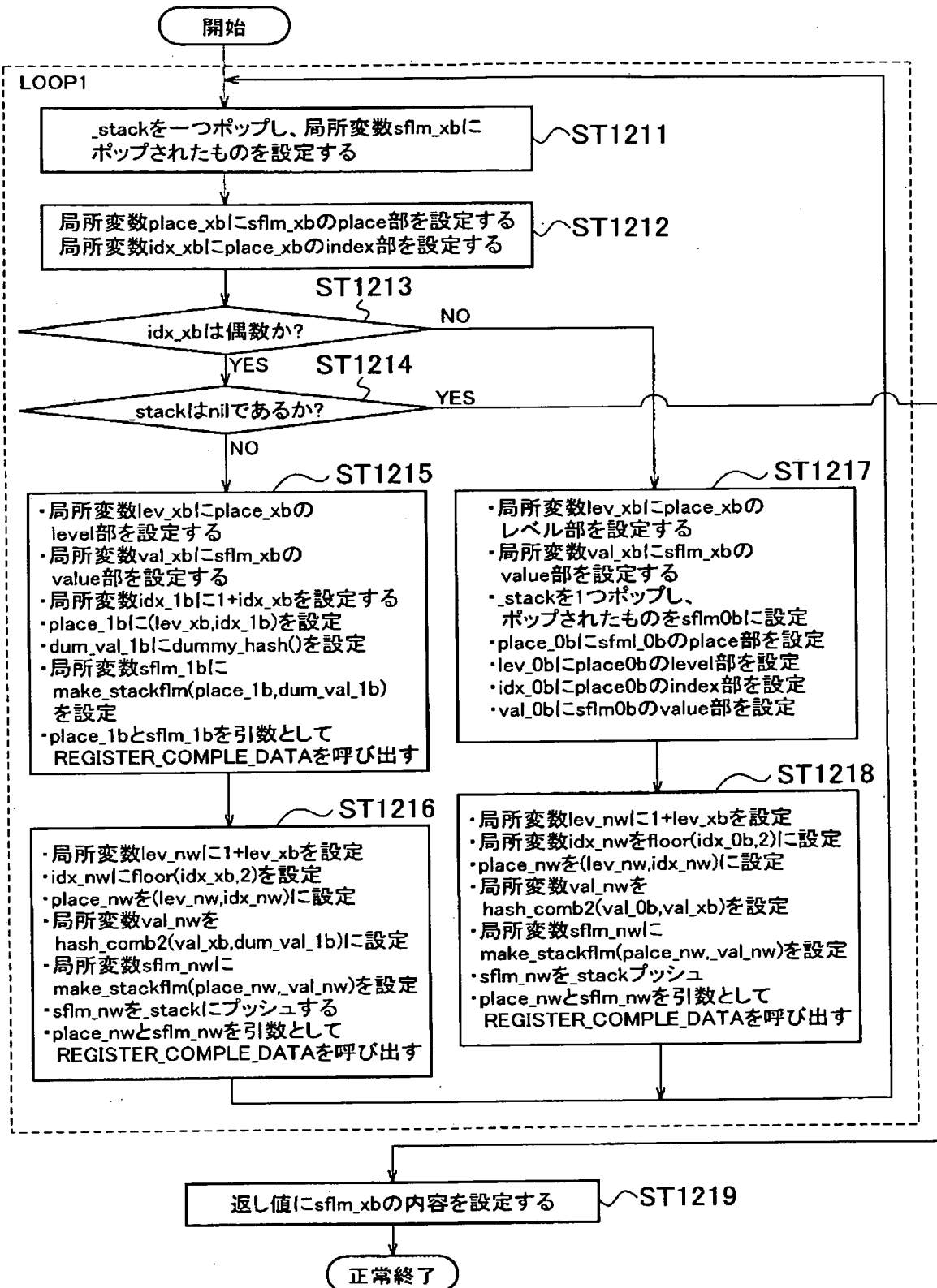
【図 20】



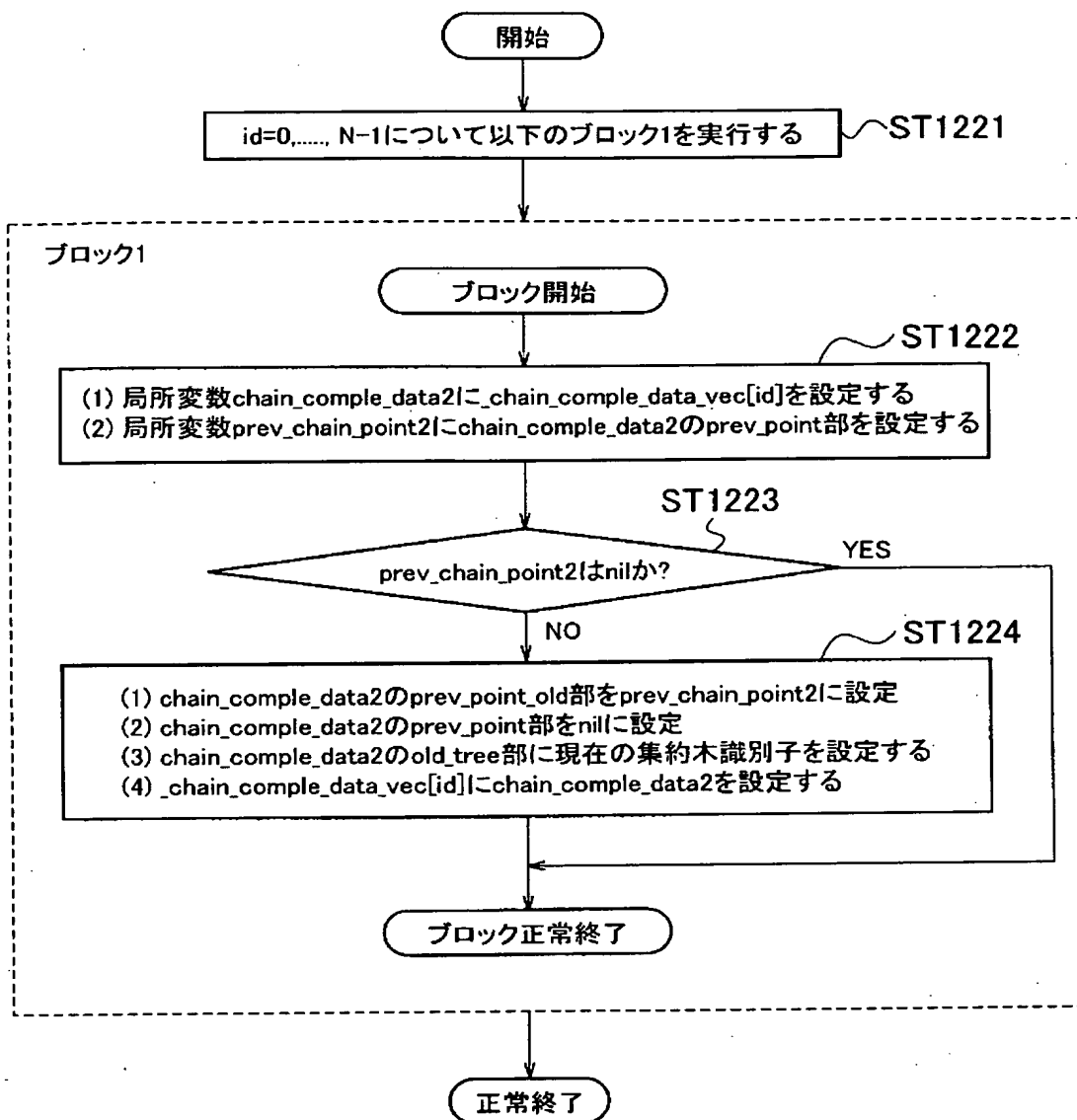
【図 21】



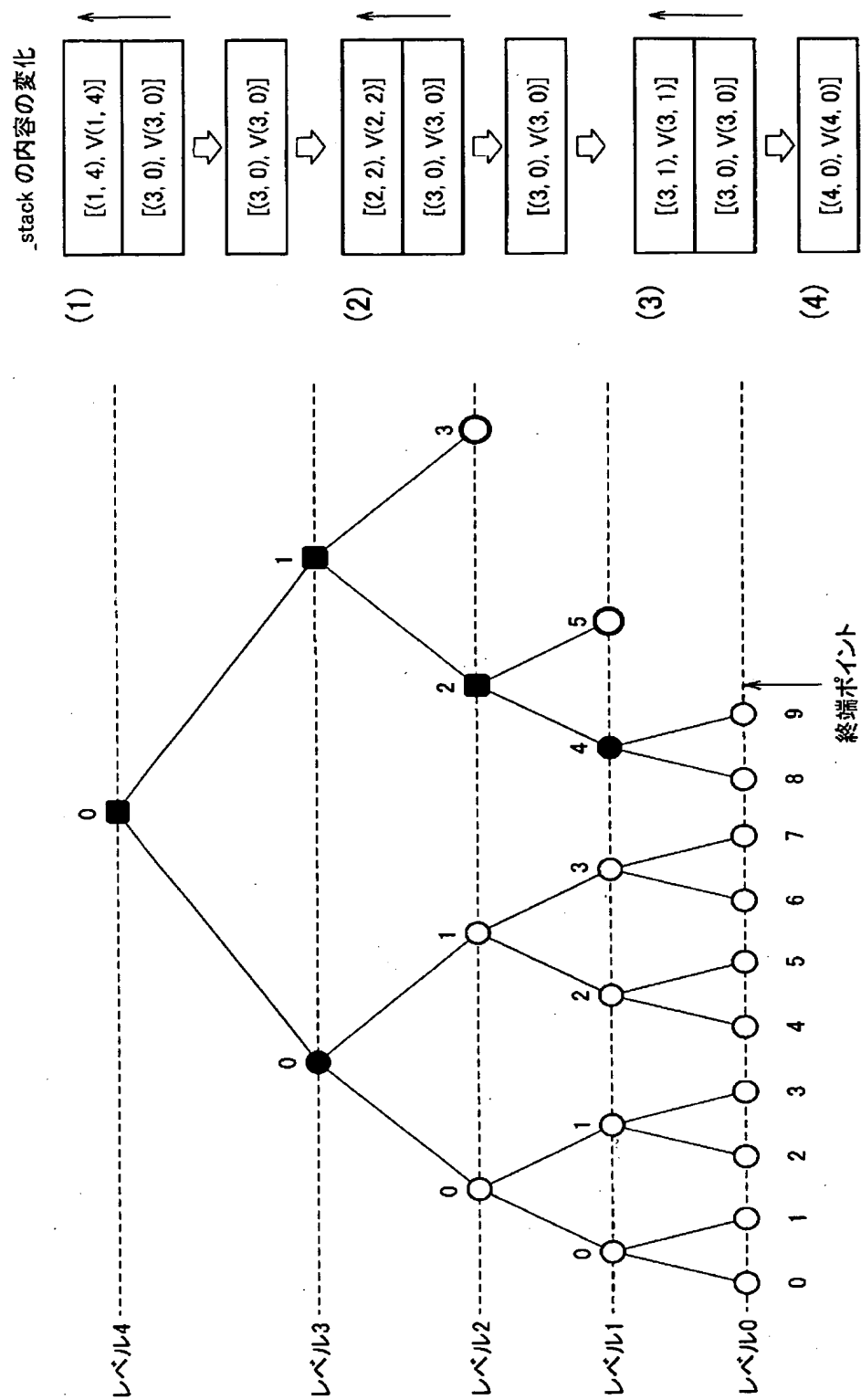
【図 2 2】

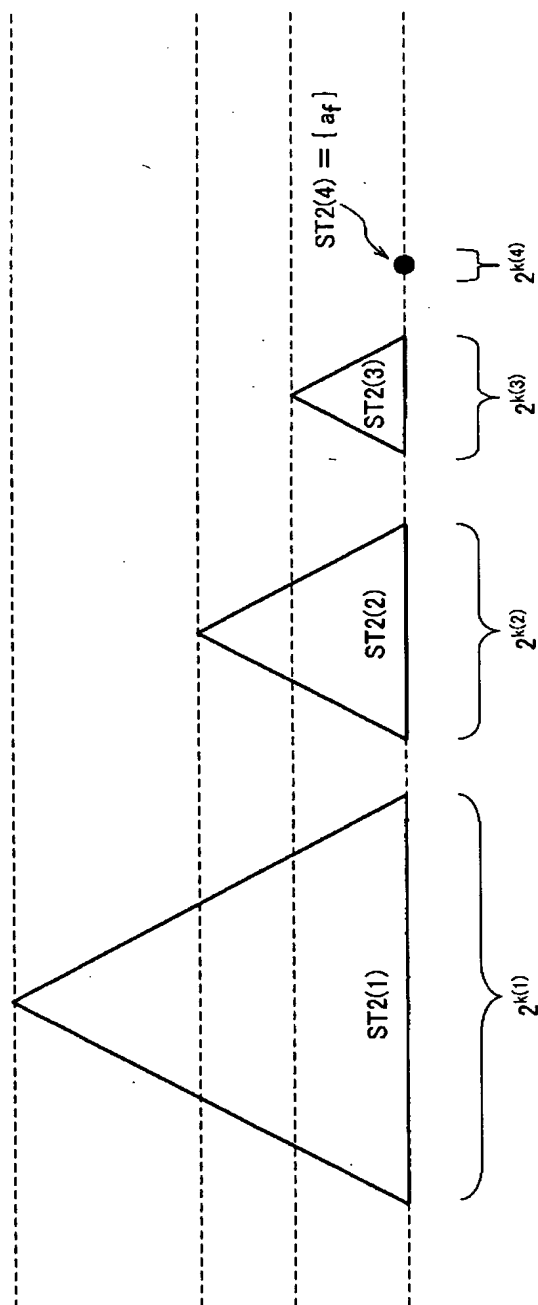


【図 2 3】

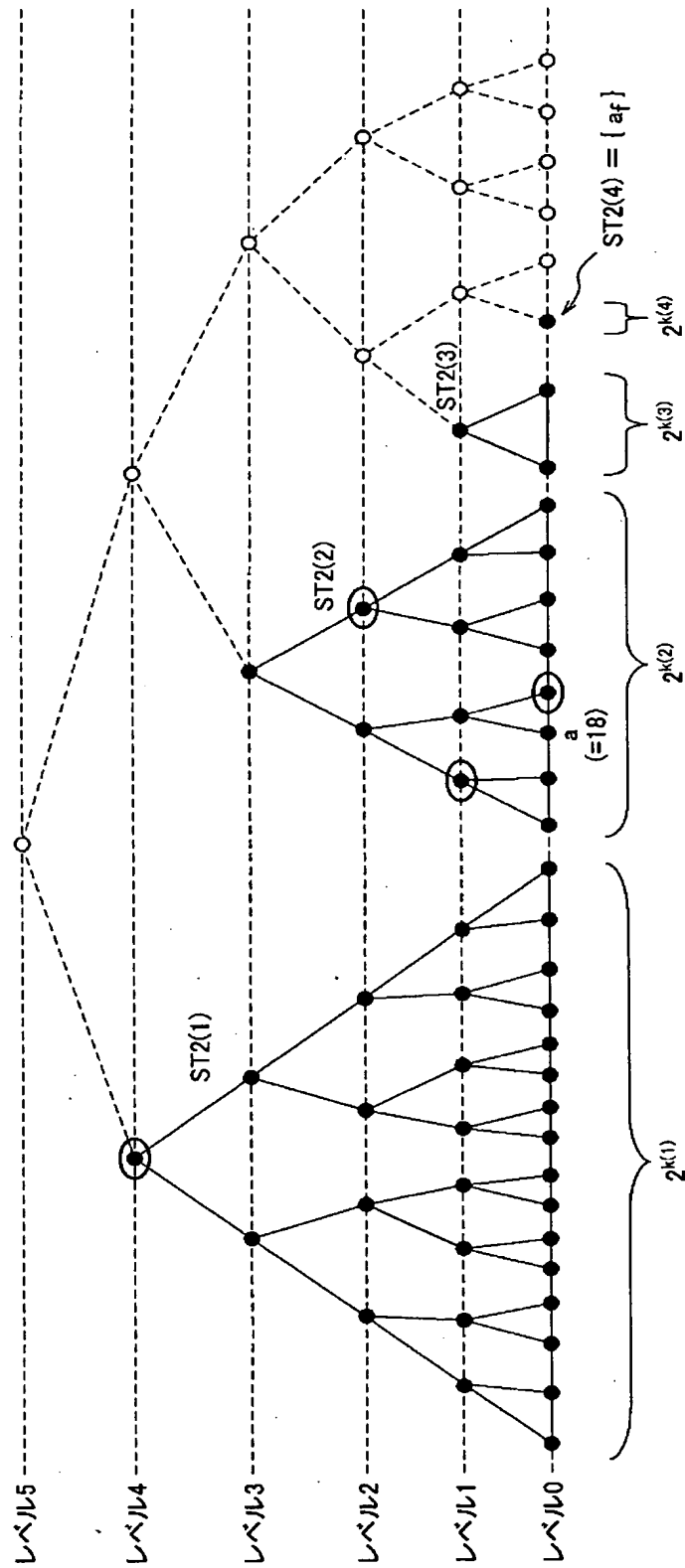




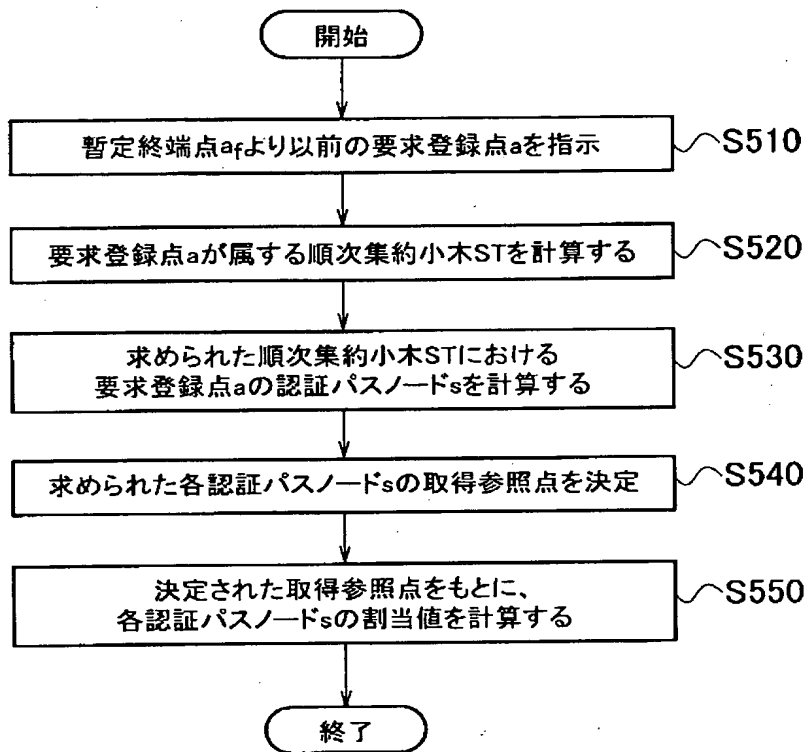


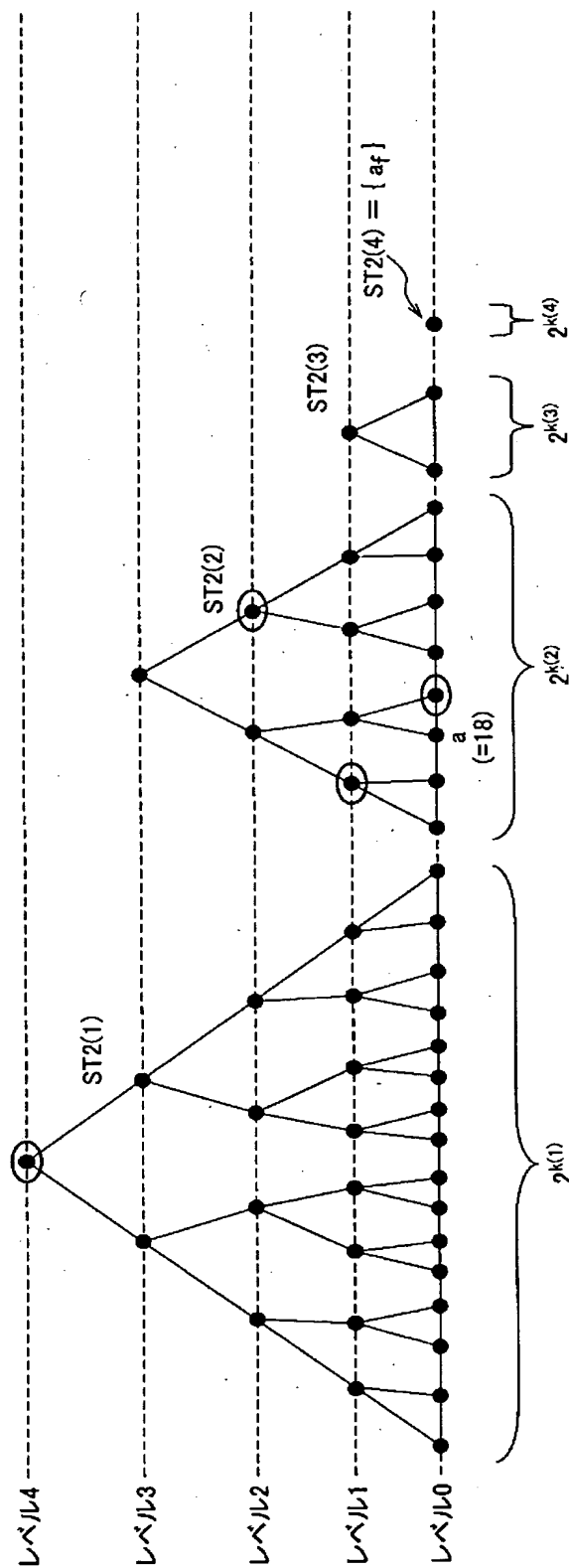


$$k(1) > k(2) > k(3) > k(4) = 0$$

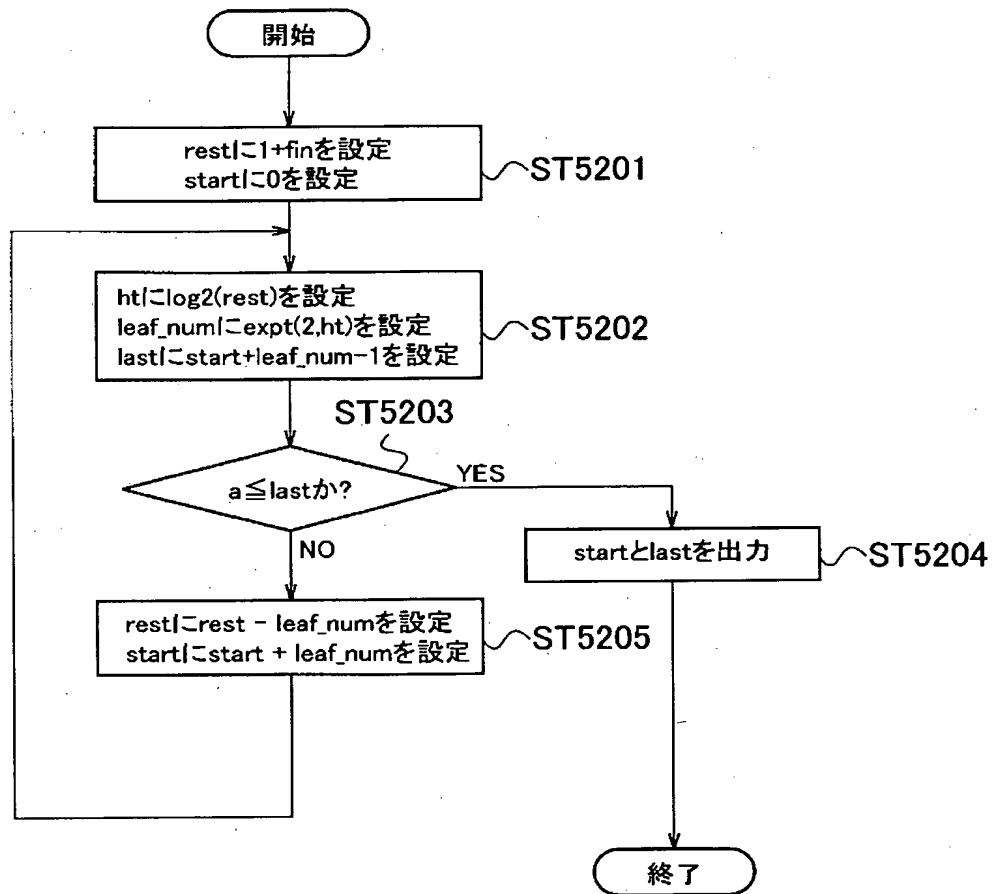


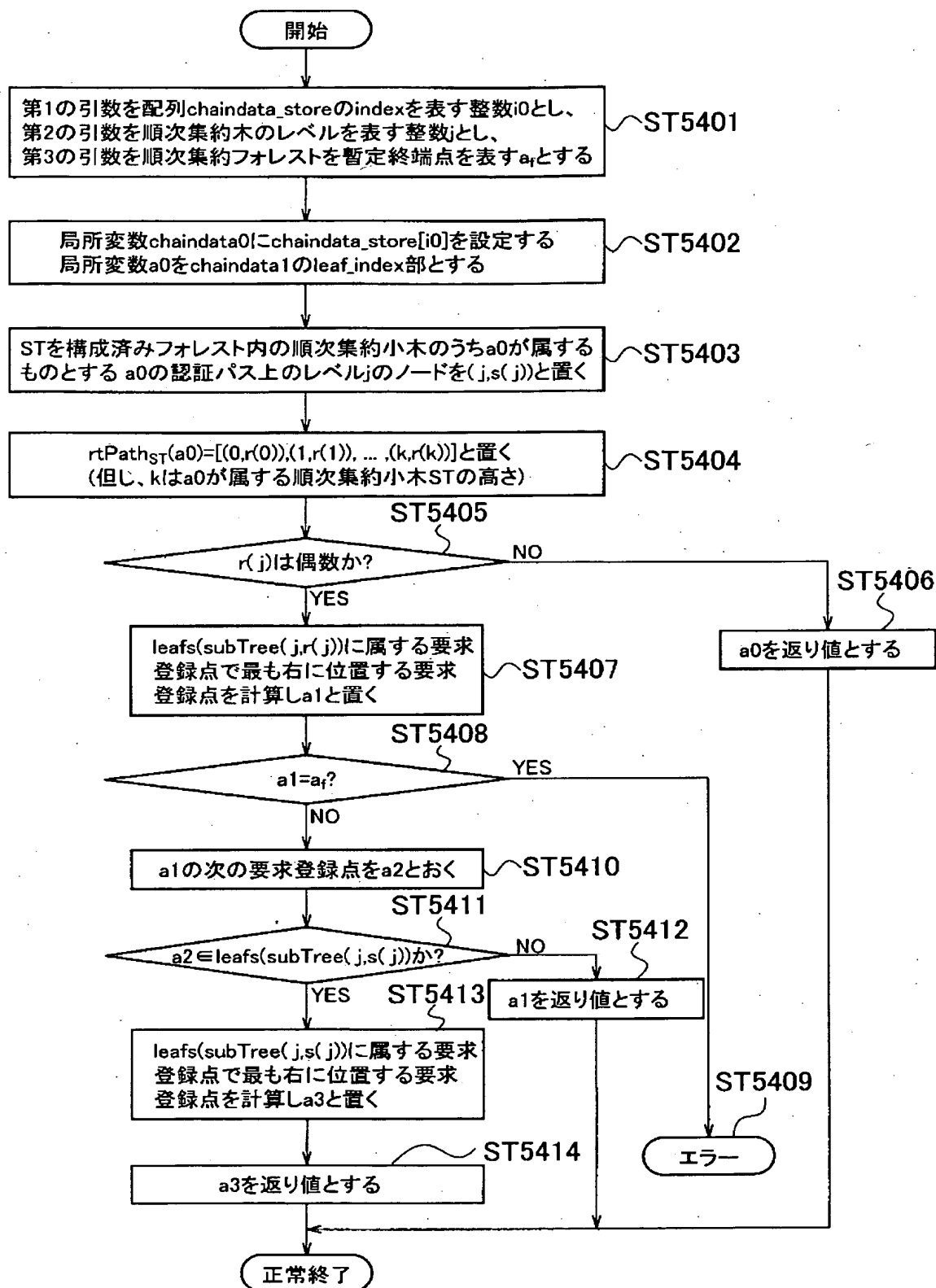
$$k(1)=4 > k(2)=3 > k(3)=1 > k(4)=0$$

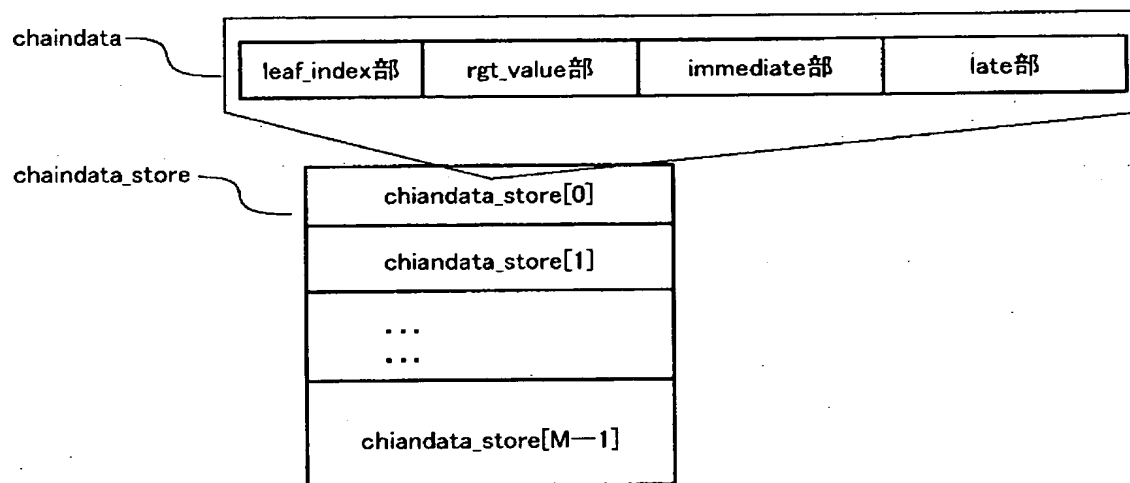




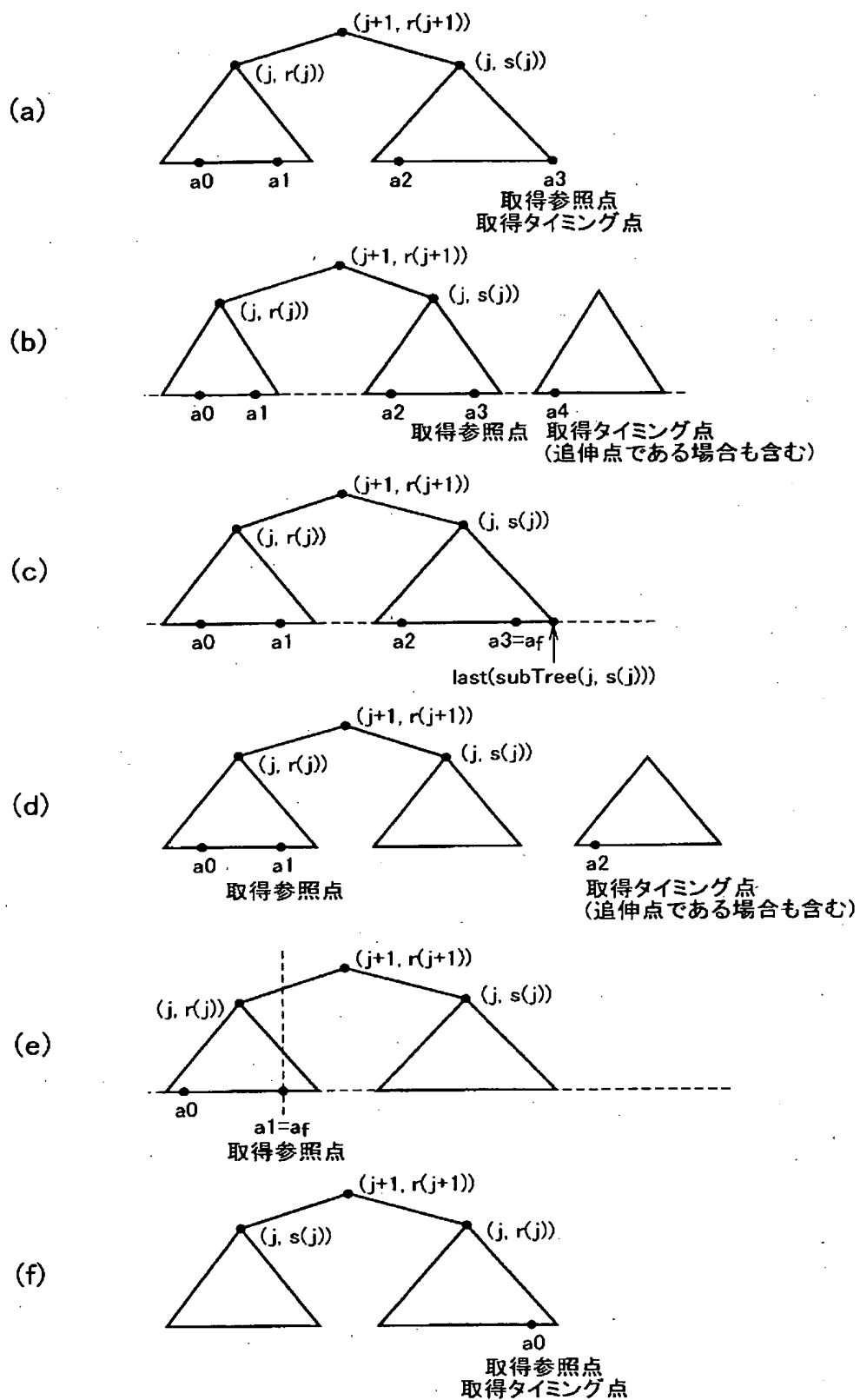
$$k(1)=4 > k(2)=3 > k(3)=1 > k(4)=0$$

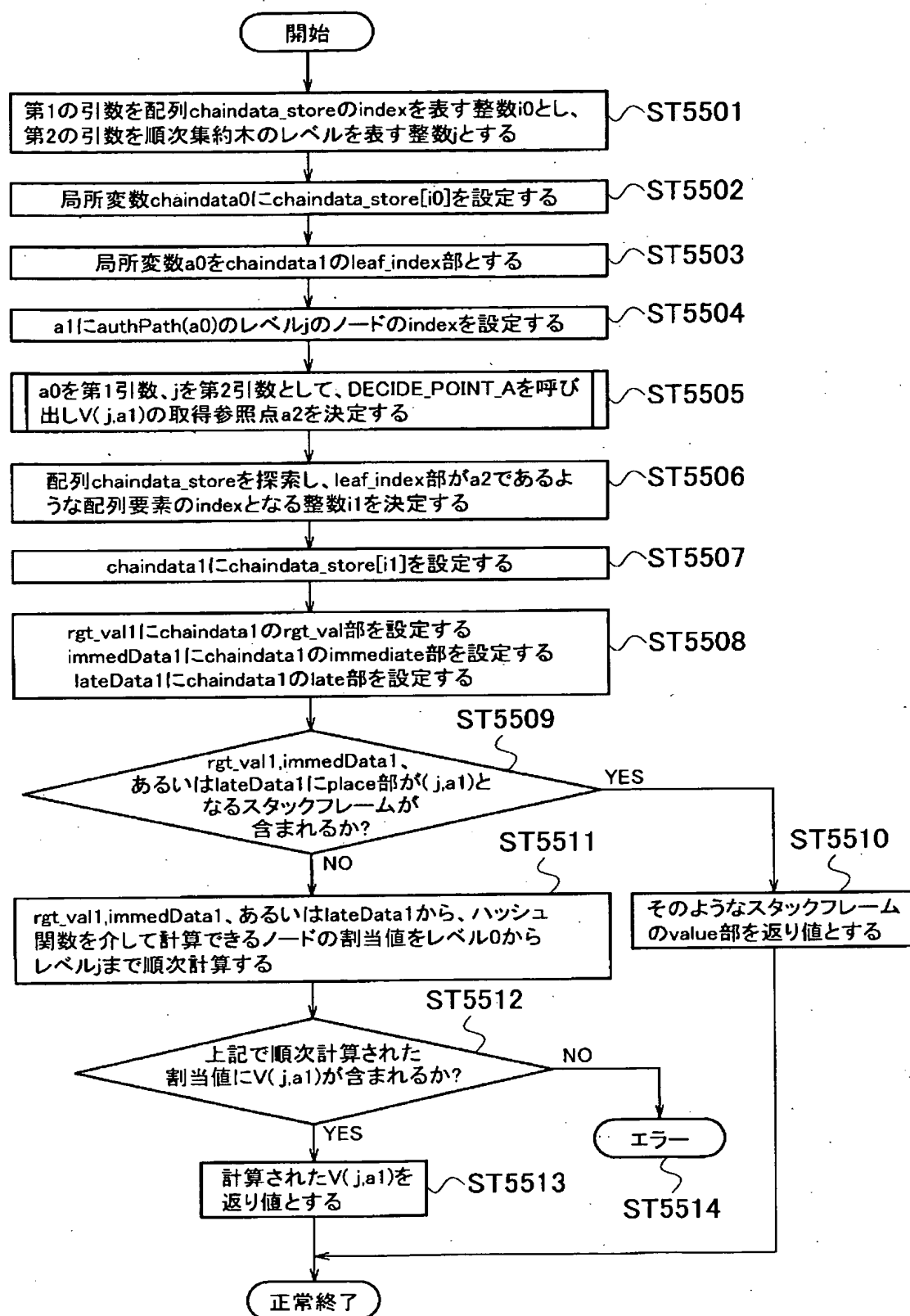


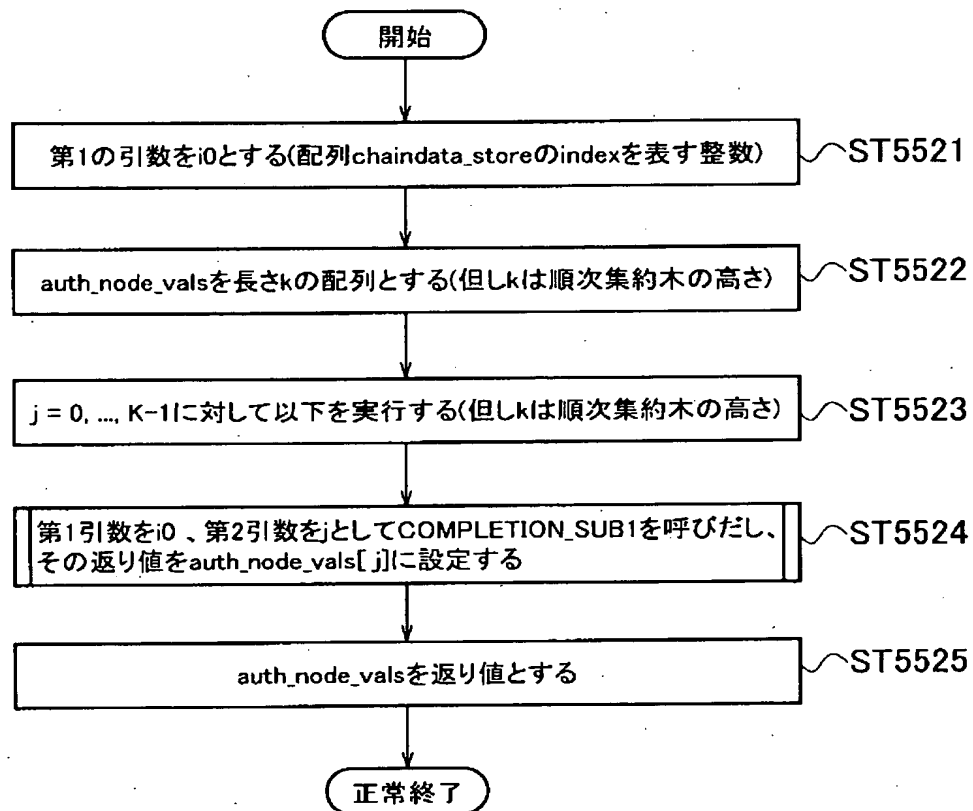


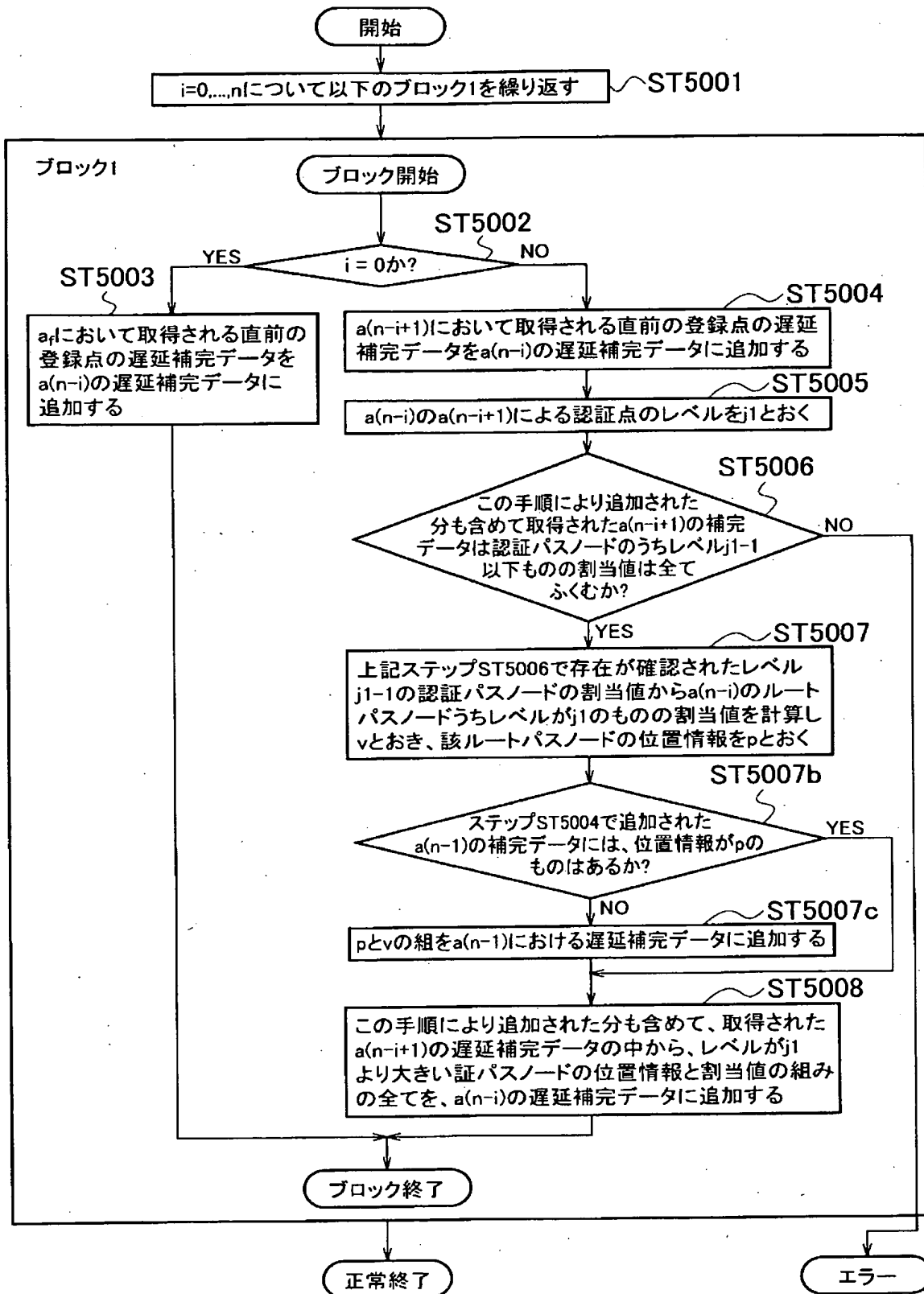




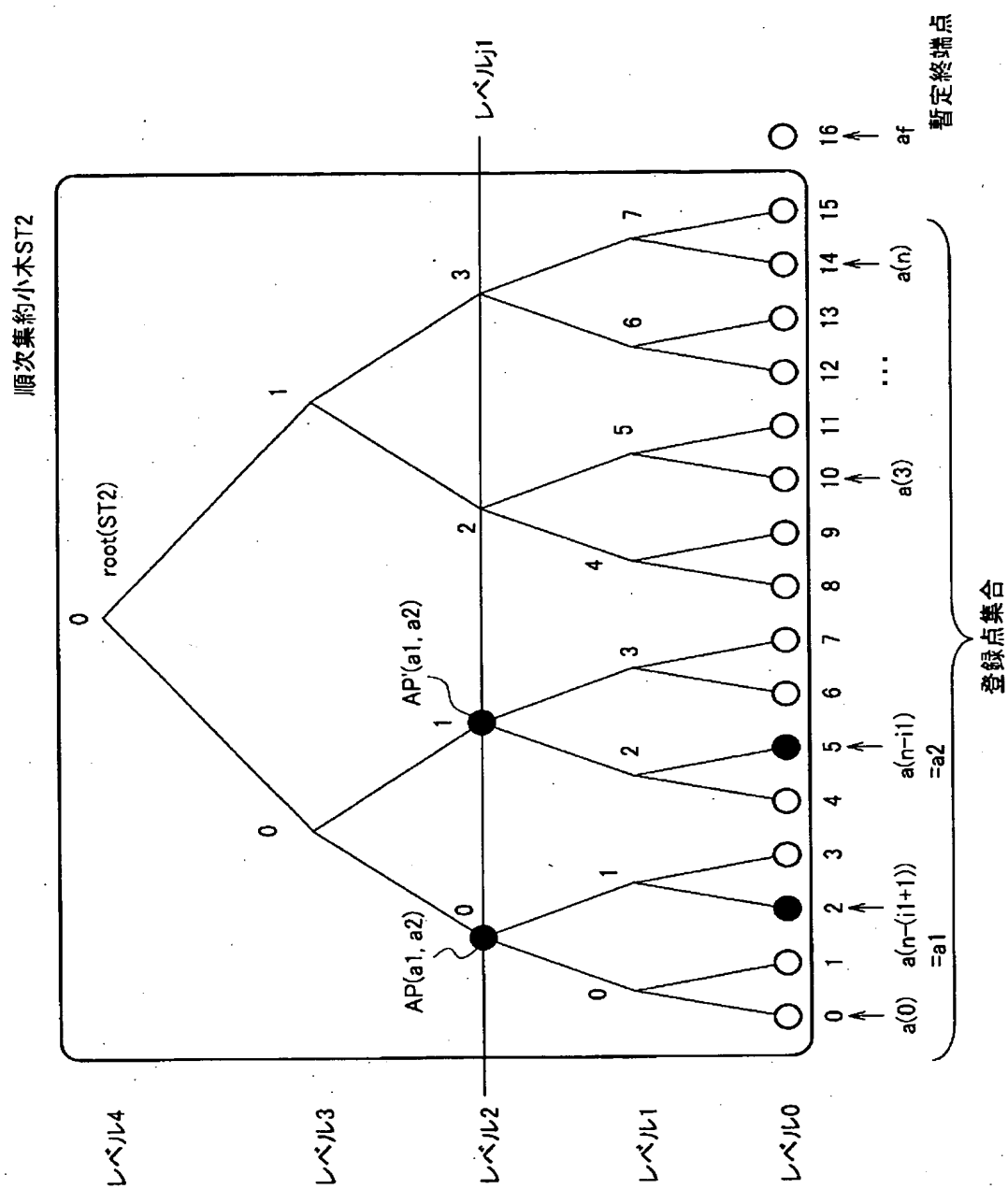


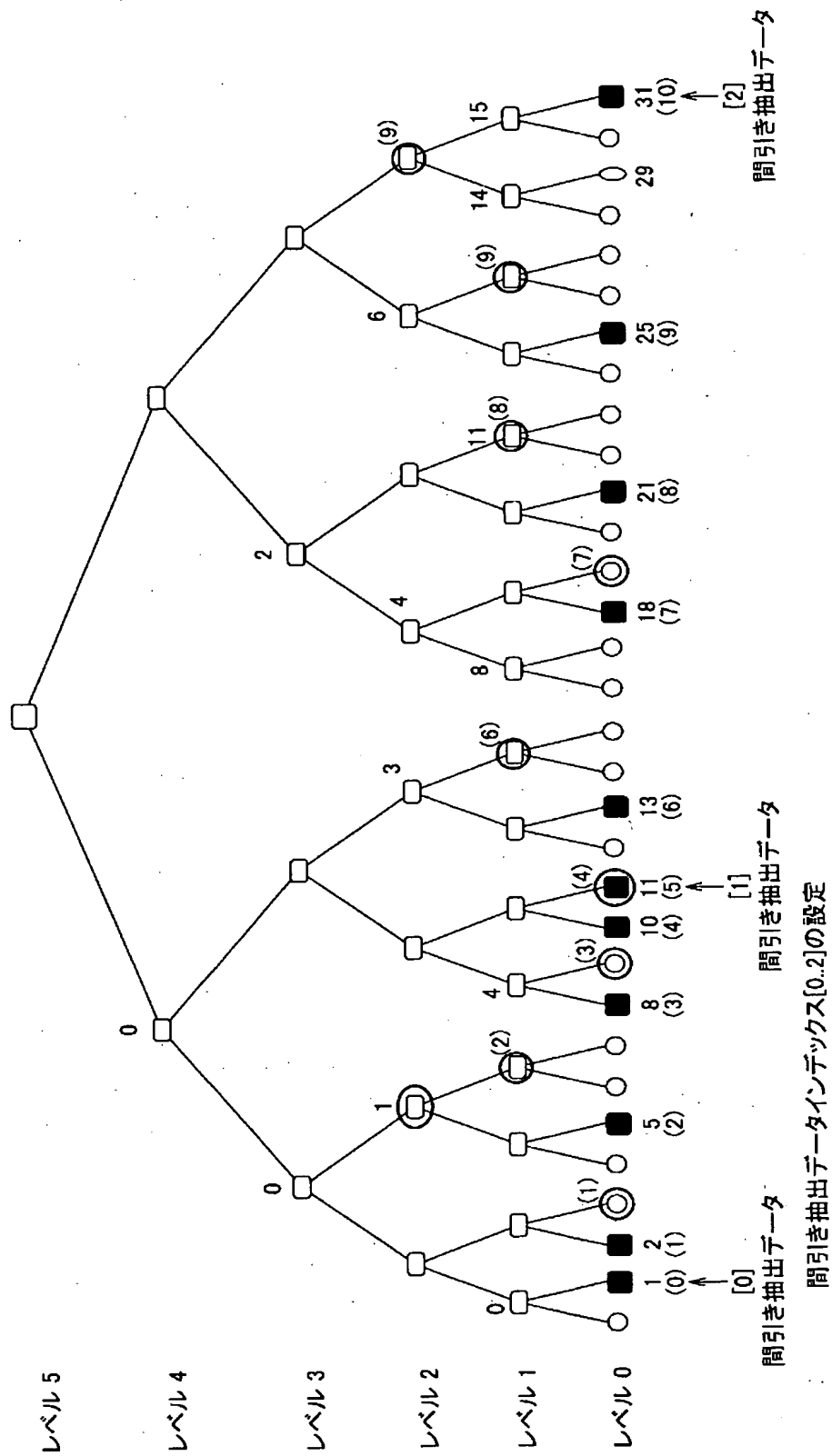


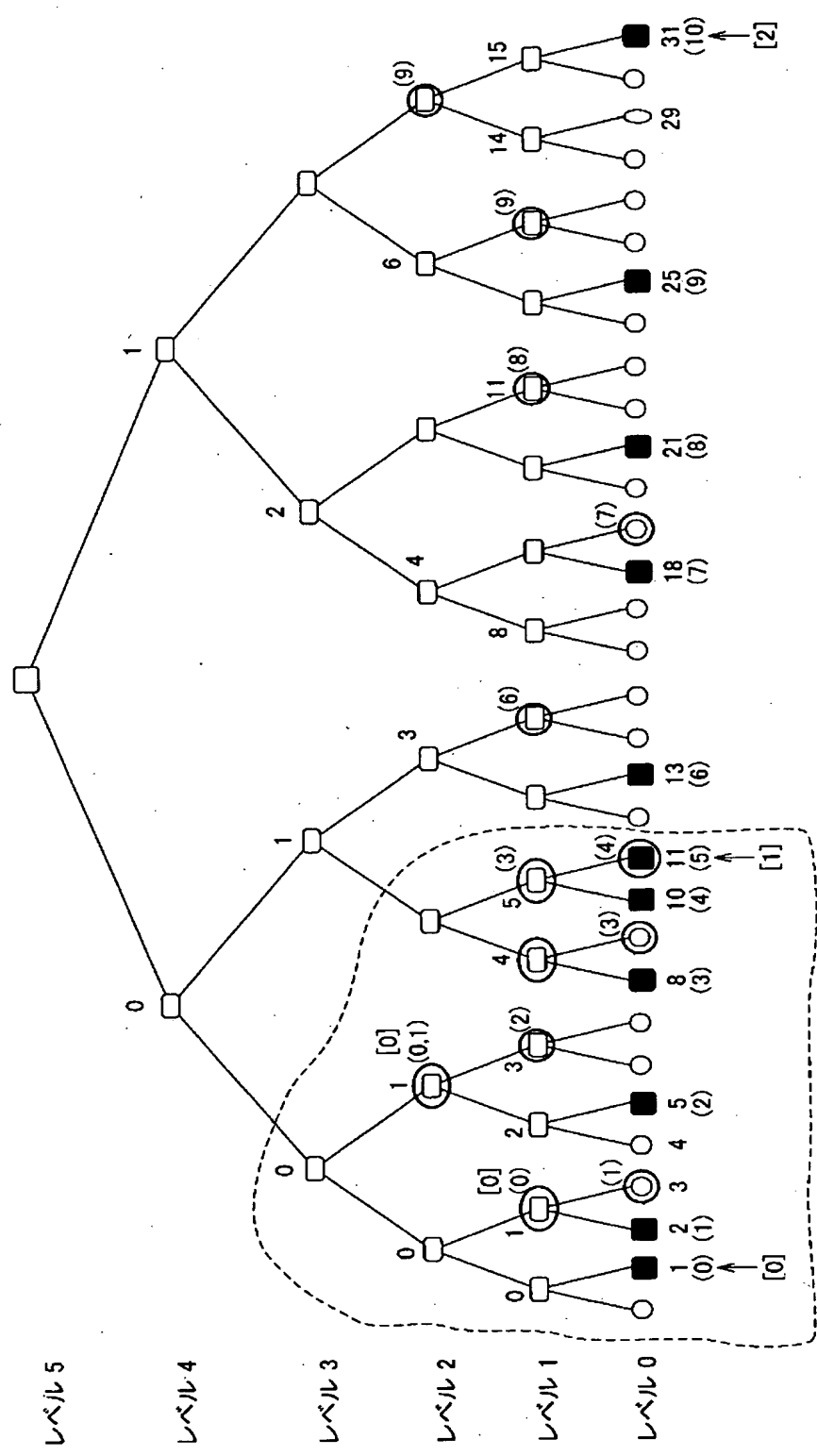






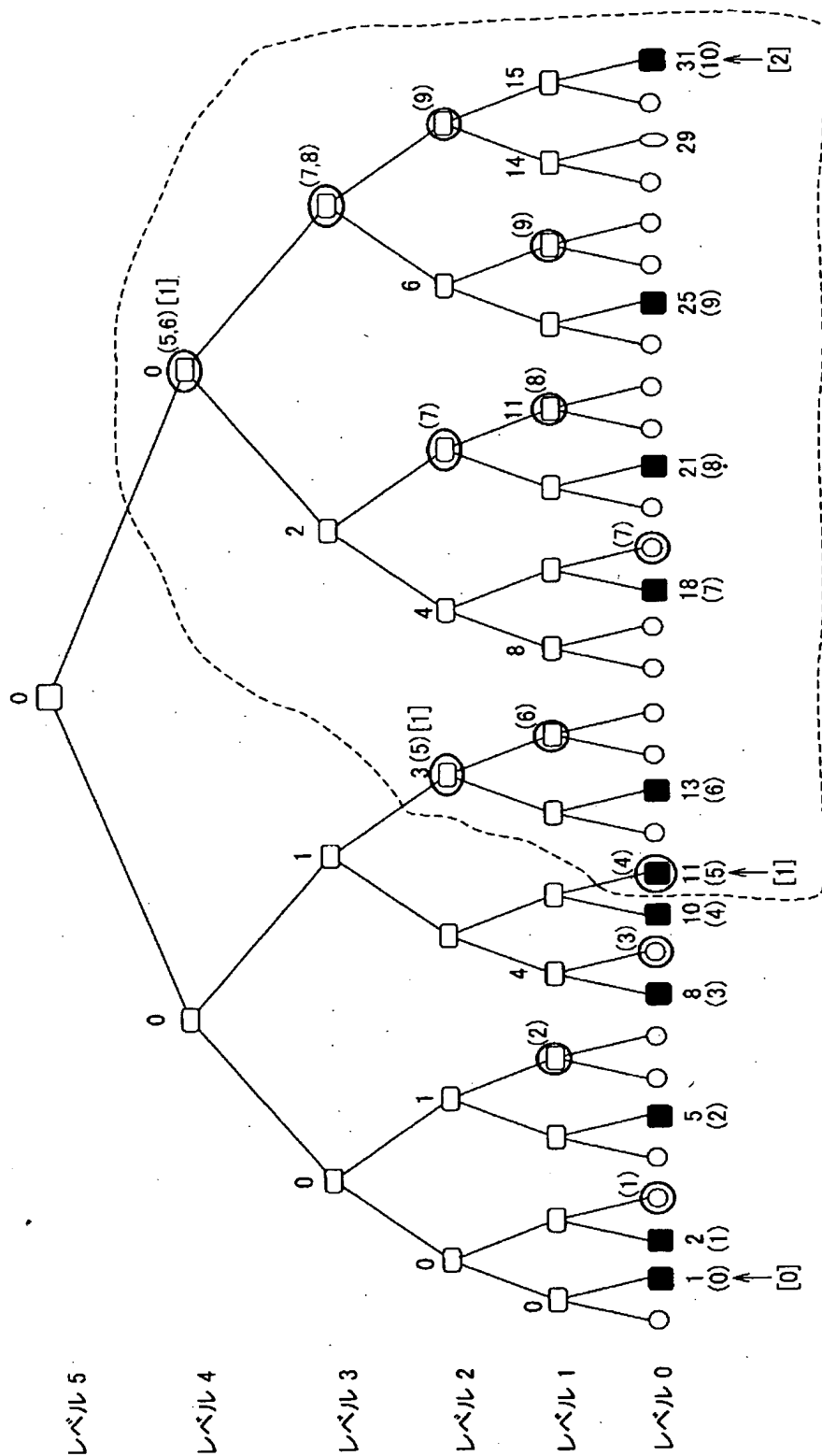






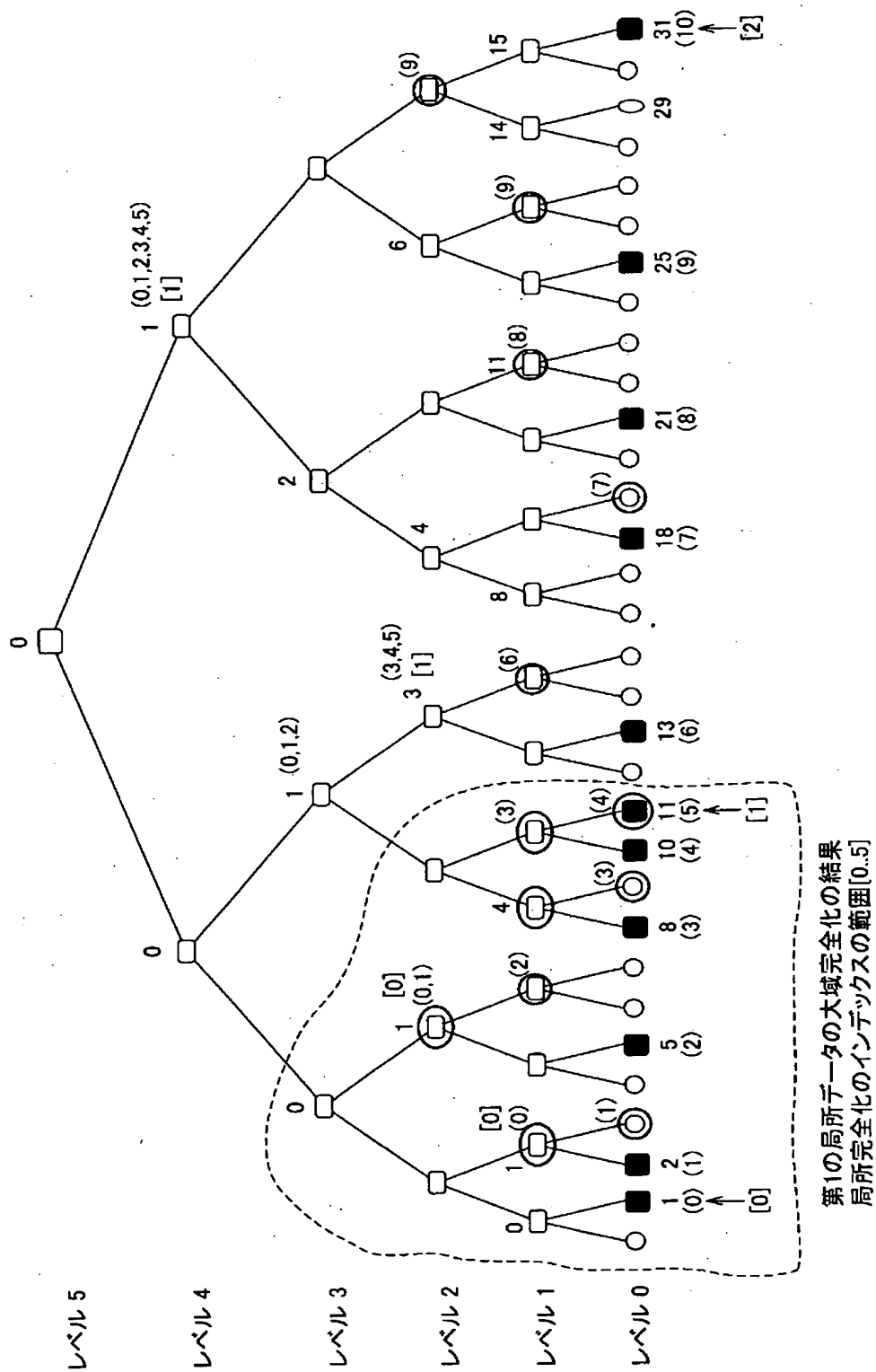
第1の局所データの局所完全化の結果  
局所完全化の元のインデックスの範囲[0..5]  
間引き抽出データインデックスの範囲[0..1]



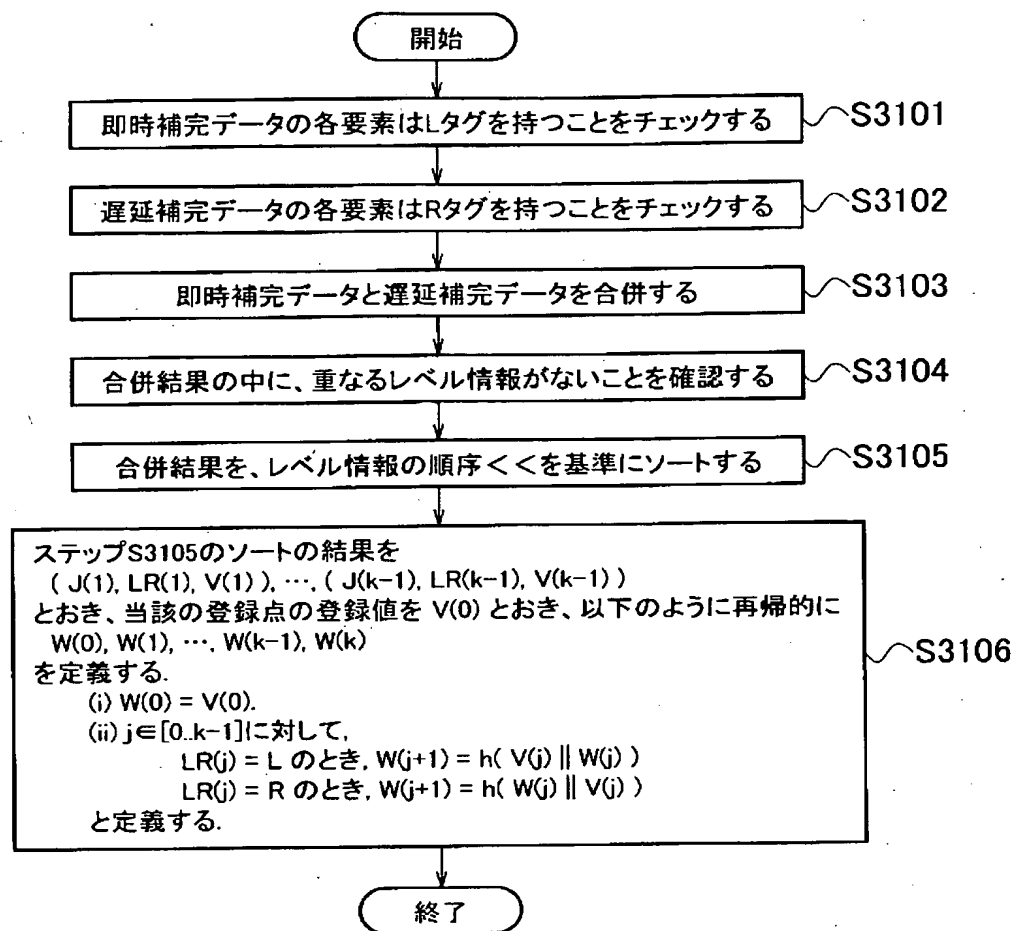


第2の局所データの局所完全化の結果  
局所完全化のインデックスの範囲[5..10]  
間引き抽出データインデックスの範囲[1..2]

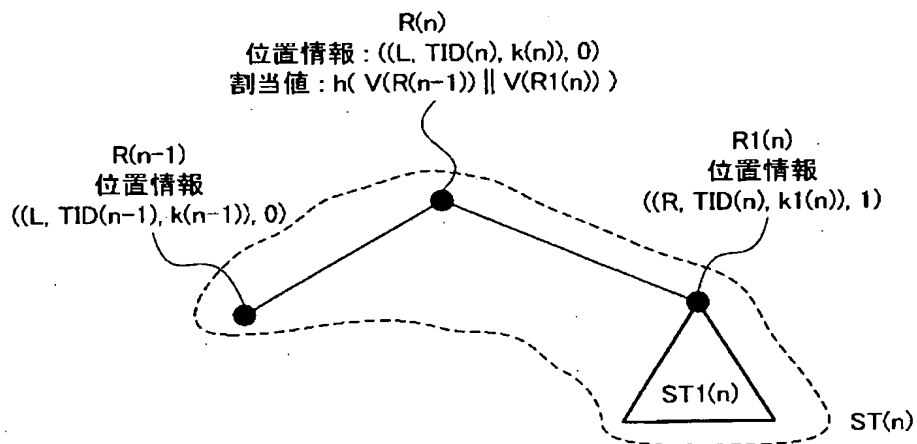


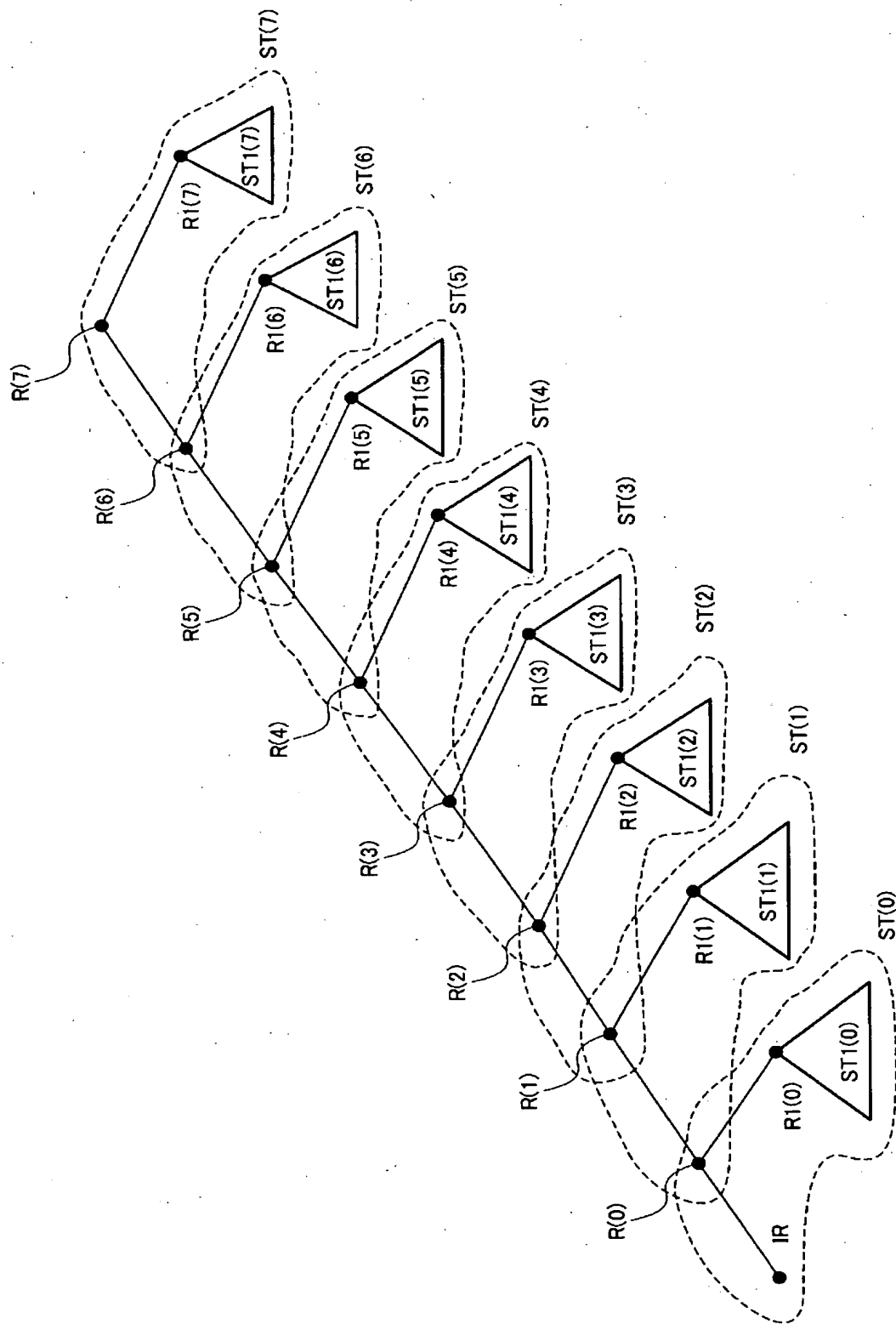


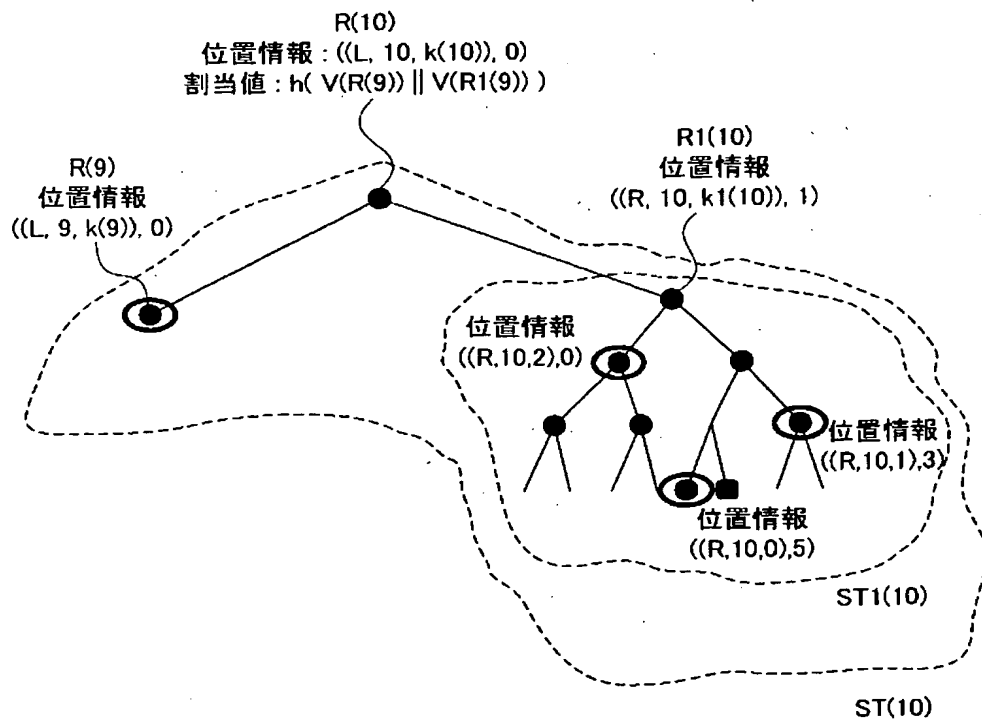
【図 4 3】



【図 4 4】

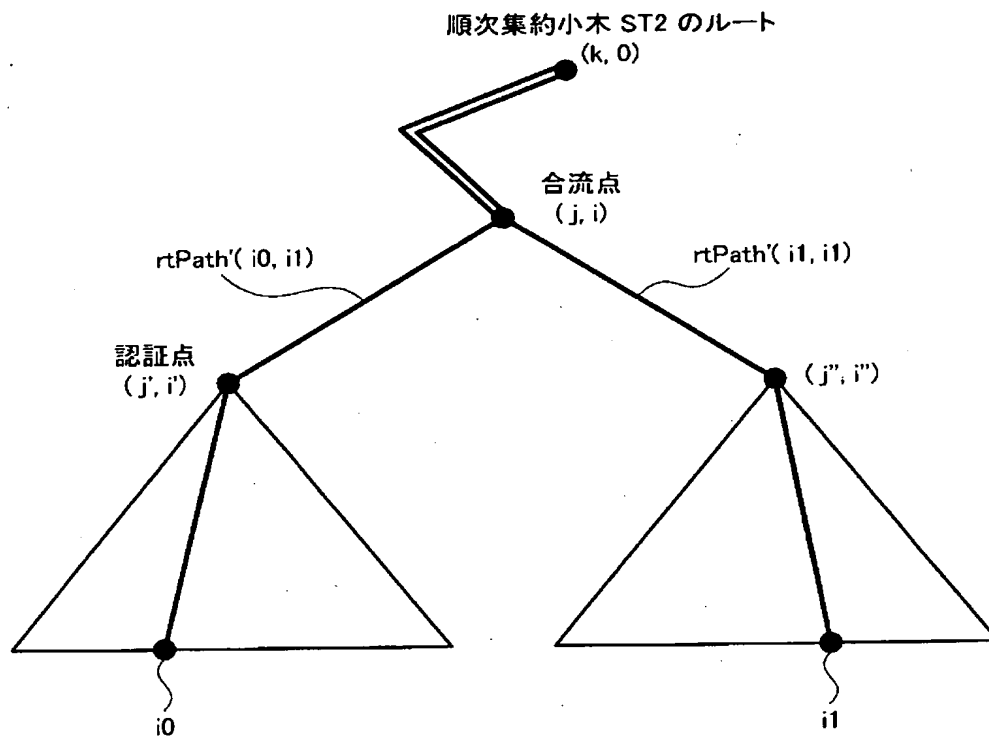




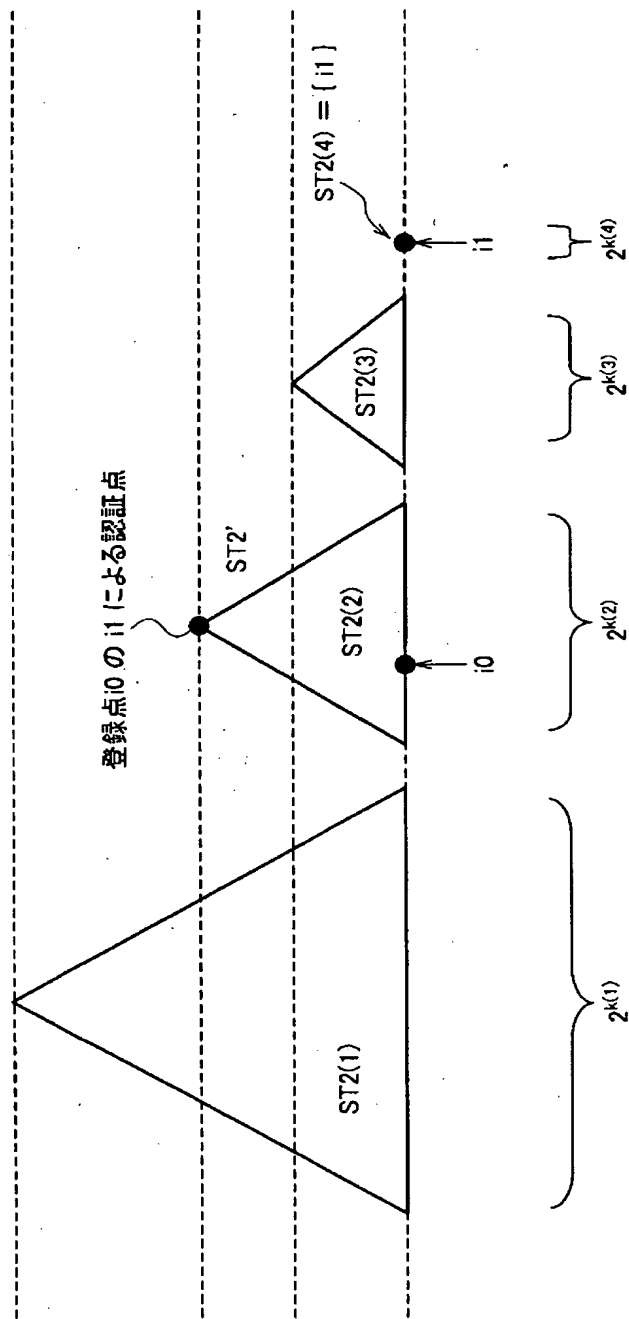


【図 4 7】

$i_0$  と  $i_1$  が  $i_1$  時点における1つの順次集約小木に属するとき



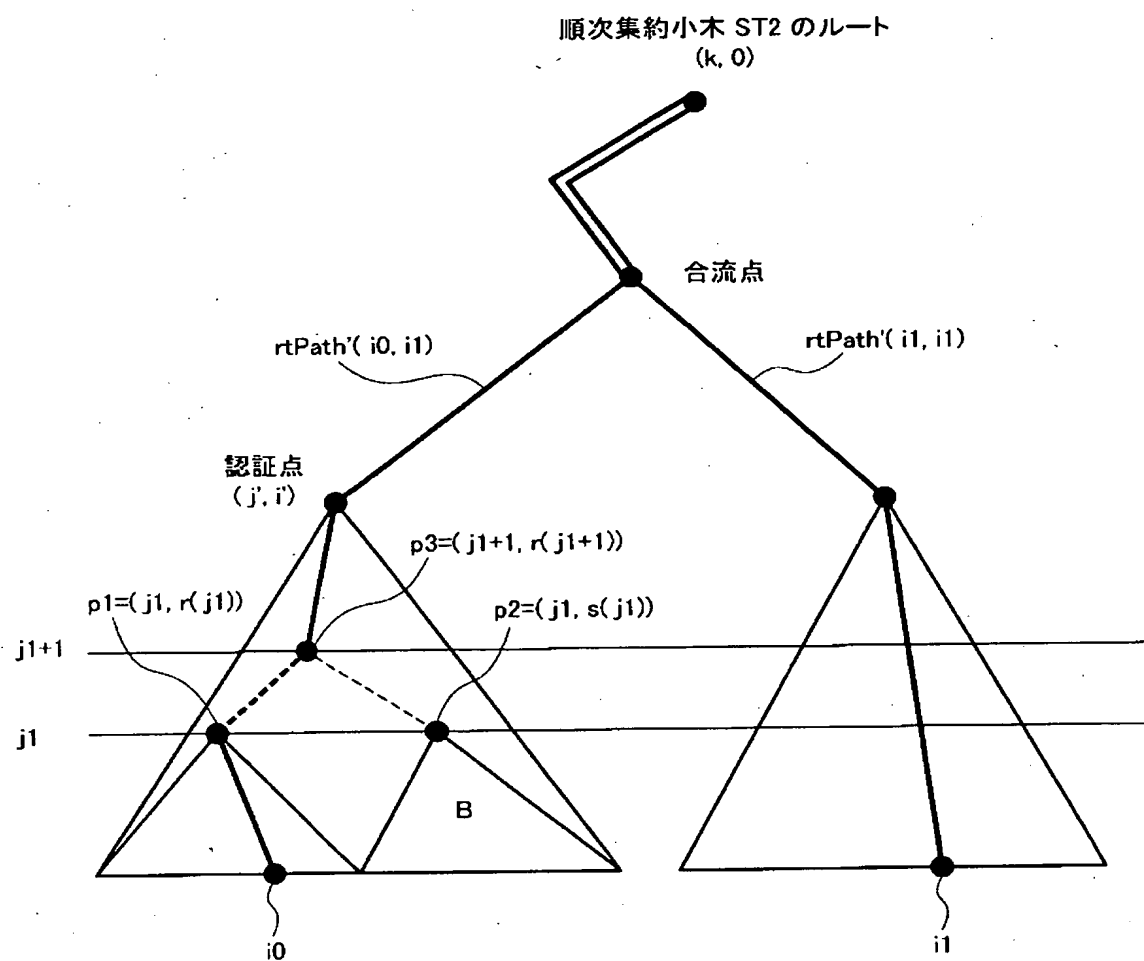
$i_0$  と  $i_1$  が  $i_1$  時点における1つの順次集約小木に属さないとき



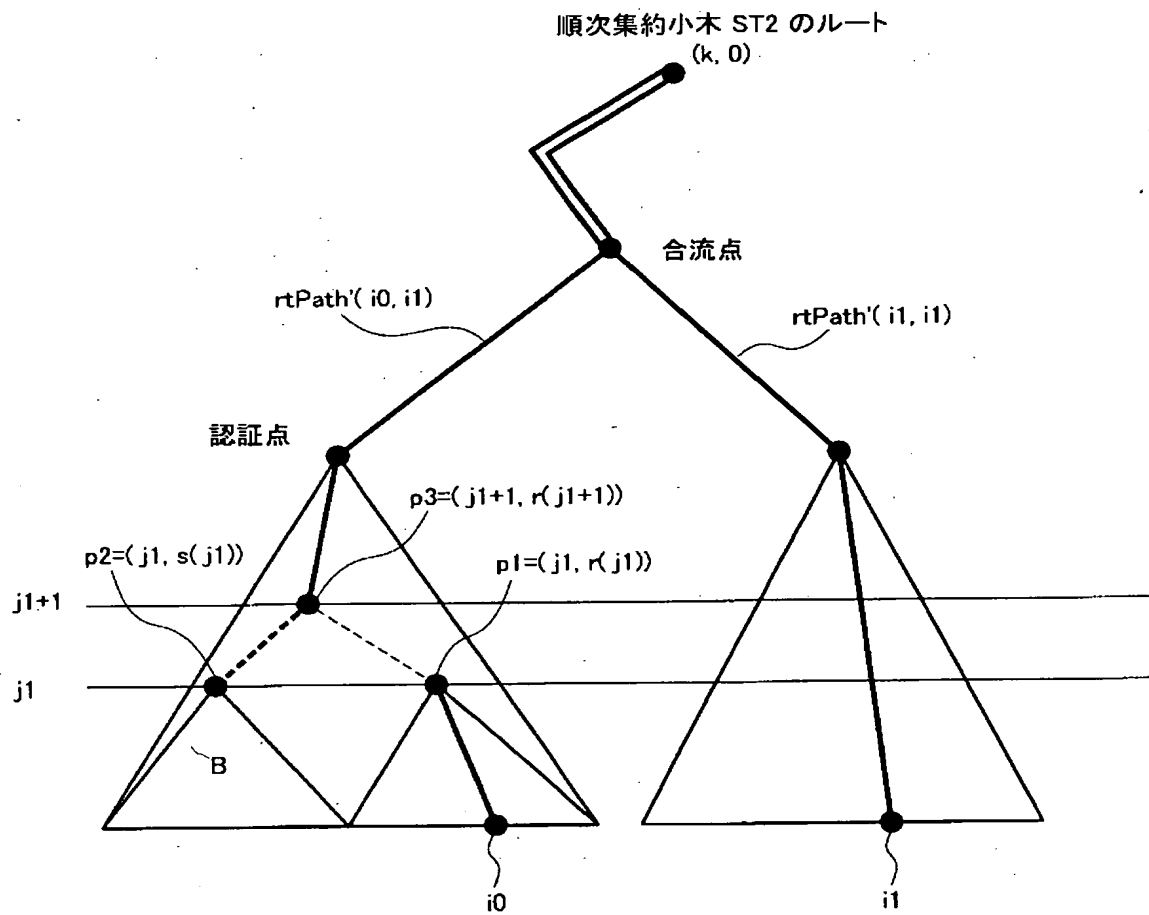
$$k(1) > k(2) > k(3) > k(4) = 0$$



$i_0$  と  $i_1$  が  $i_1$  時点における1つの順次集約小木に属するとき

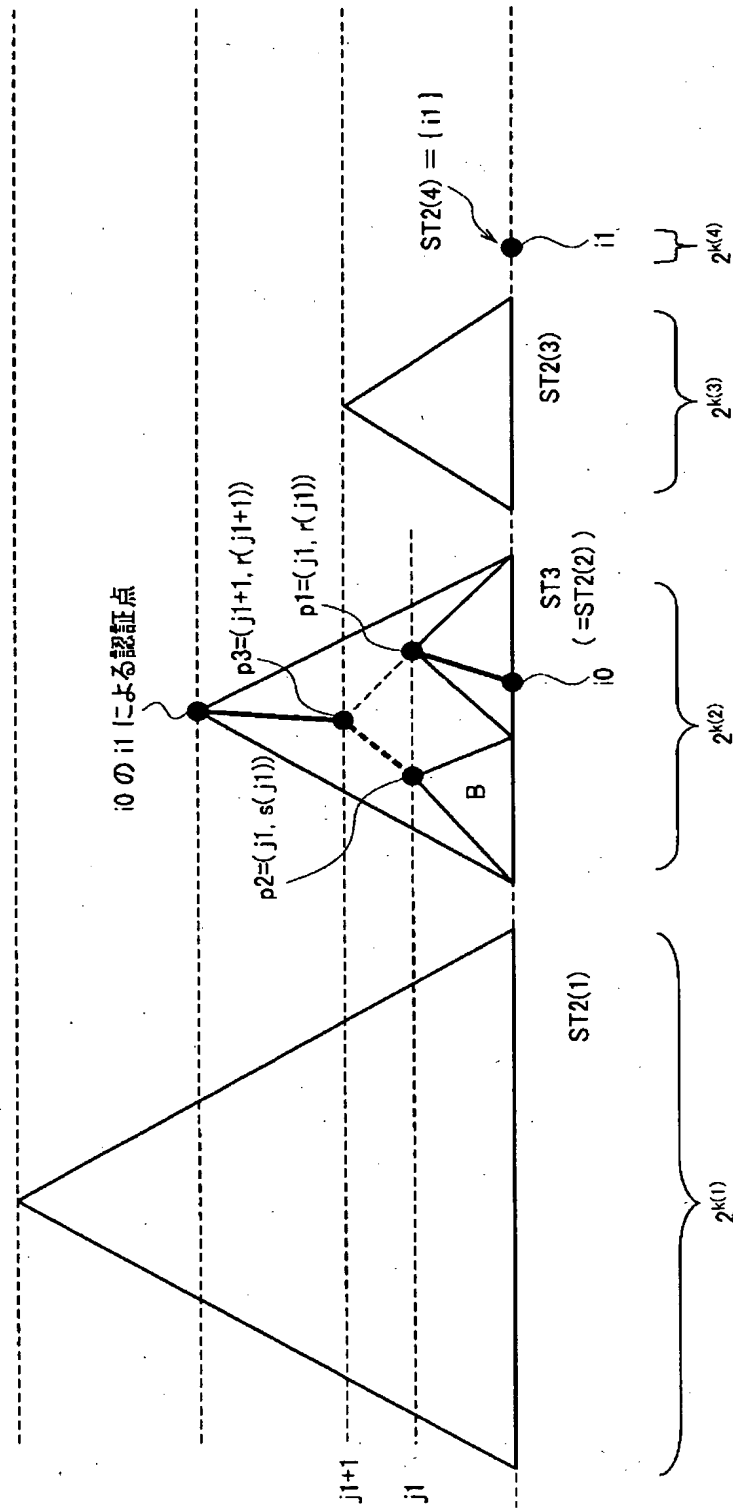


$i_0$  と  $i_1$  が  $i_1$  時点における1つの順次集約小木に属するとき

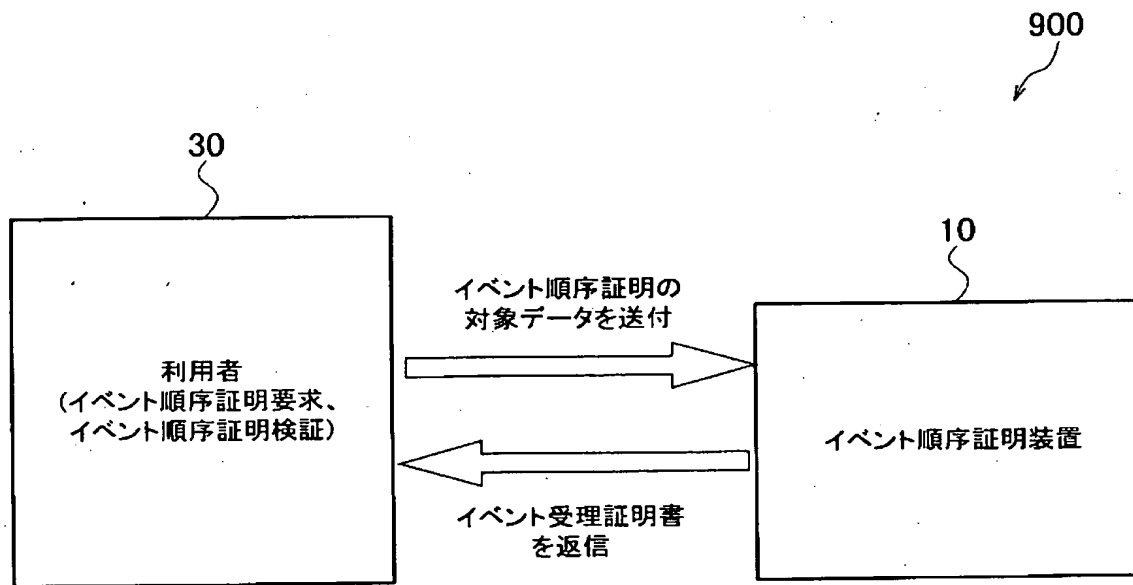




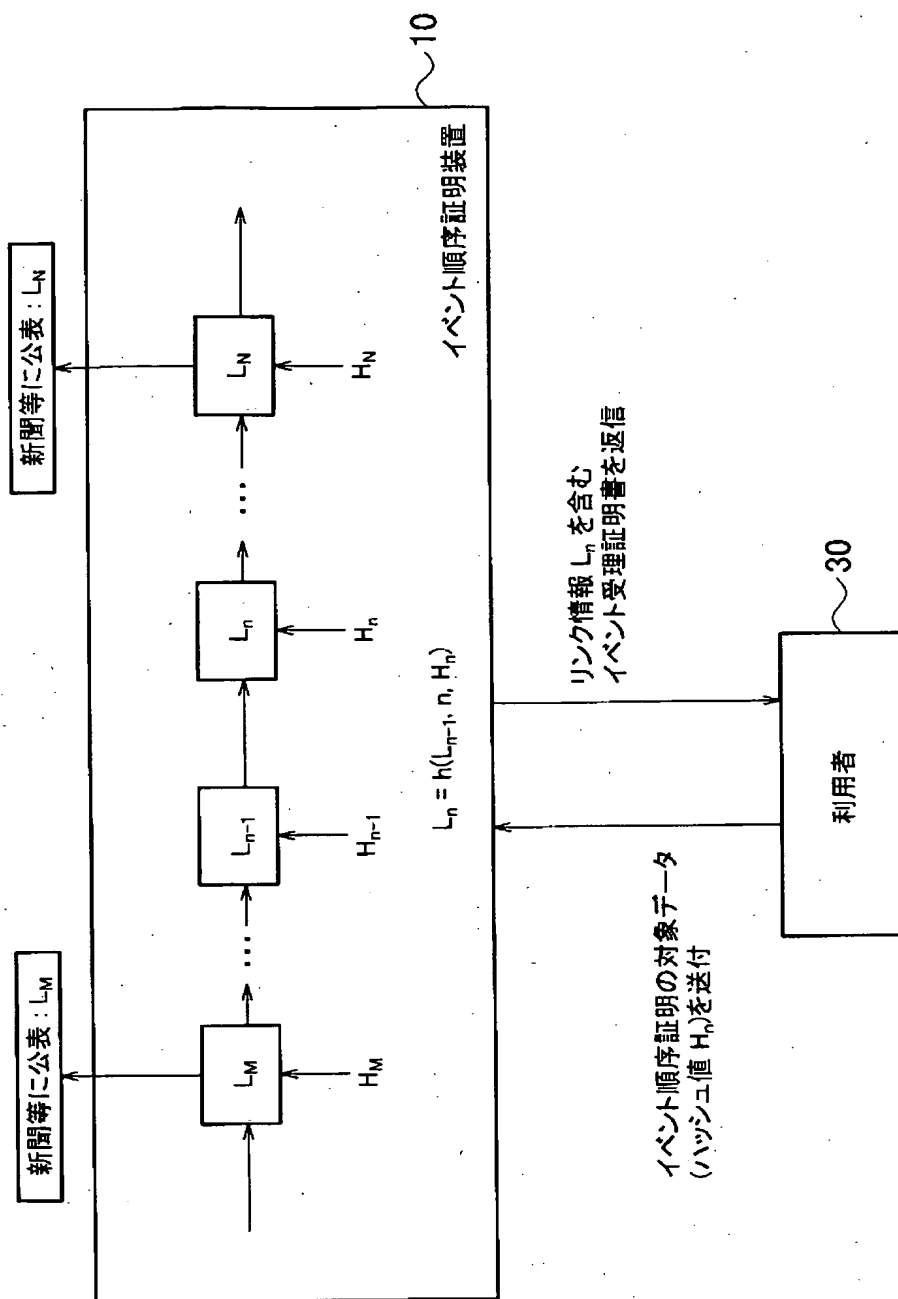
$i_0$  と  $i_1$  が  $j$  時点における1つの順次集約小木に属さないとき



$$k(1) > k(2) > k(3) > k(4) = 0$$



910



【書類名】要約書

【要約】

【課題】本構造を用いてイベント順序を証明するイベント順序証明システムにおいて、イベント順序証明要求をまとめた公表データを用いなくても、証明機関から発行されたイベント順序受理証明書の検証を行うことができる。

【解決手段】イベント順序証明システム100は、イベント順序証明装置1、複数のイベント順序証明利用者装置21、及び、以上の各装置を相互に接続するコンピュータネットワーク3を備えており、証明装置1が利用者装置21からのイベント順序証明要求に応じて、イベント順序受理証明書を含むイベント順序証明応答を利用者装置21に返信する。ここで、イベント順序証明応答は、シーケンス補完方式（登録点の即時補完データ及び該登録点より前に登録された各登録点の登録点における遅延補完データを証明応答に含む）による証明応答であり、利用者装置21は、証明装置1から受け取ったこの複数の証明応答から受理証明書の正当性を検証する。

【選択図】図1

出願人履歴

000004226

19990715

住所変更

591029286

東京都千代田区大手町二丁目3番1号

日本電信電話株式会社